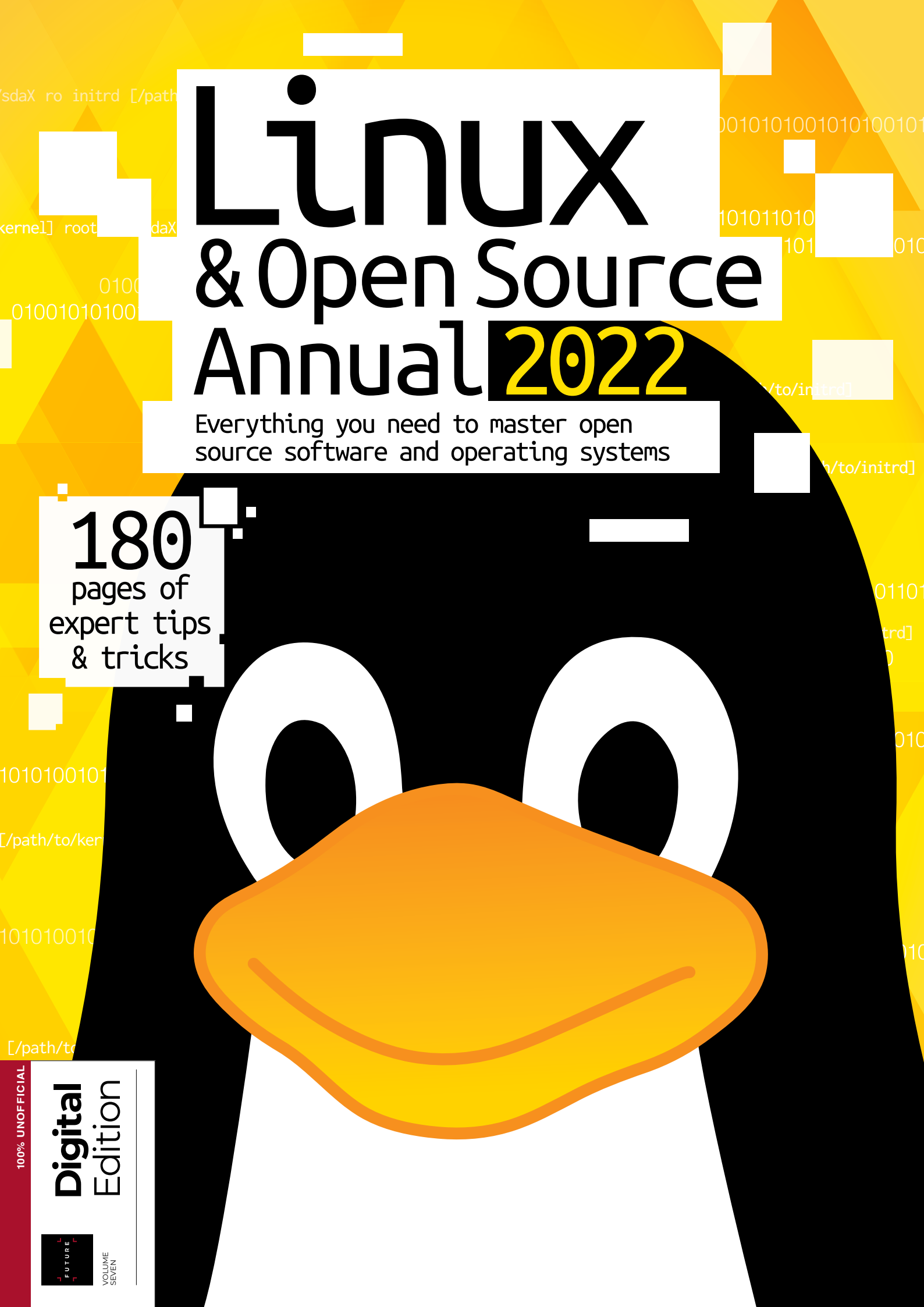
Linux & Open Source Annual 2022

Everything you need to master open source software and operating systems

180 pages of expert tips & tricks



100% UNOFFICIAL

Digital Edition

FUTURE

VOLUME SEVEN

Welcome to

Linux & Open Source Annual 2022

Free and Open Source Software (FOSS) has become an international phenomenon in recent years. The FOSS philosophy of protecting user freedoms sets it apart from many commercial software providers. And by constantly updating and improving, it pushes the boundaries of what can be achieved with software, allowing you freedom in other ways: to create, to hack (in the tinkering sense; behave yourselves!), to explore new ways of doing things. We've crammed as much exciting information as we possibly can into this annual, including the hottest distros and FOSS for a wide range of fun and practical applications. This collection truly is an abundance of Linux and open source knowledge, ready for you to jump right into.

So, what are you waiting for?

This bookazine is printed on recycled paper. It's important that we care about our planet and make a difference where we can, for us and every generation that follows.

Linux & Open Source Annual 2022

Future PLC Quay House, The Ambury, Bath, BA1 1UA

Linux & Open Source Annual 2022 Editorial
Compiled by **Drew Sleep & Briony Duguid**
Senior Art Editor **Andy Downes**
Head of Art & Design **Greg Whitaker**
Editorial Director **Jon White**

Linux Format Editorial
Editor **Neil Mohr**
Art Editor **Efrain Hernandez-Mendoza**
Group Editor in Chief **Graham Barlow**
Senior Art Editor **Jo Gulliver**

Contributors
Jonni Bidwell, Shashank Sharma, Keith Edmunds, Ed Bennett, John Knight, Mike Bedford, Aaron Peters, Kent Elchuk, Alexander Tolstoy

Photography
All copyrights and trademarks are recognised and respected

Advertising
Media packs are available on request
Commercial Director **Clare Dove**

International
Head of Print Licensing **Rachel Shaw**
licensing@futurenet.com
www.futurecontenthub.com

Circulation
Head of Newstrade **Tim Mathers**

Production
Head of Production **Mark Constance**
Production Project Manager **Matthew Eglinton**
Advertising Production Manager **Joanne Crosby**
Digital Editions Controller **Jason Hudson**
Production Managers **Keely Miller, Nola Cokely, Vivienne Calvert, Fran Twentyman**

Printed by William Gibbons, 26 Planetary Road, Willenhall, West Midlands, WV13 3XT

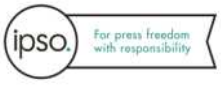
Distributed by Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU
www.marketforce.co.uk Tel: 0203 787 9001

Linux & Open Source Annual Vol 7 (TCB4047)

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
www.futureplc.com

Chief executive **Zillah Byng-Thorne**
Non-executive chairman **Richard Huntingford**
Chief financial officer **Penny Ladkin-Brand**

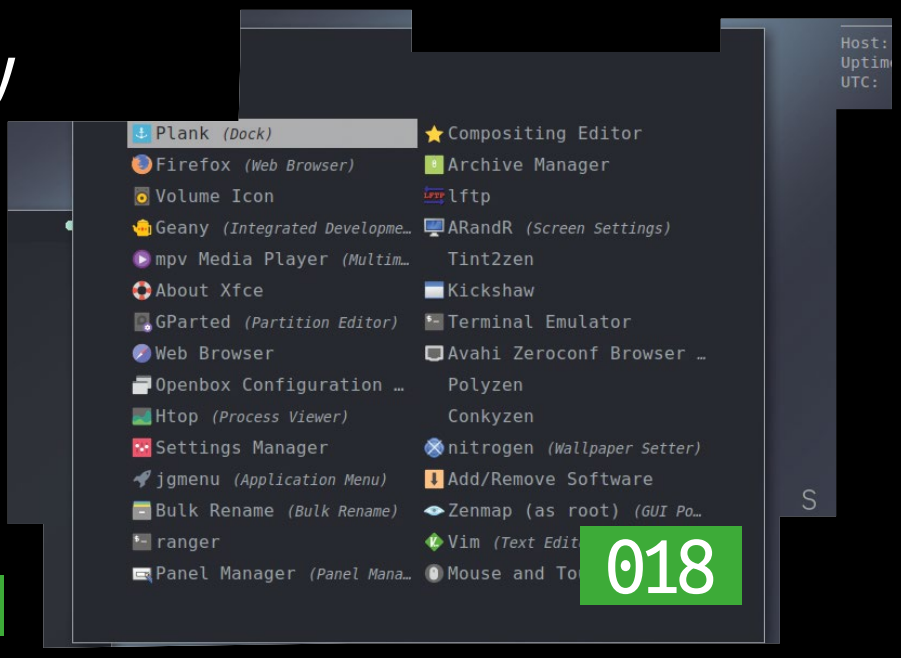
Tel +44 (0)1225 442 244



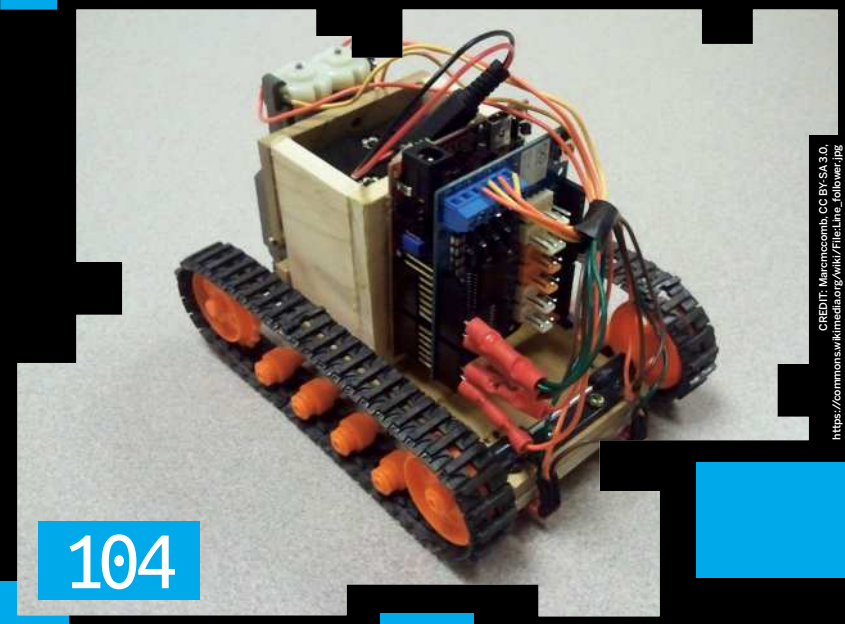
Contents

Distros

"A peculiarity you'll encounter is there's no single one Linux"

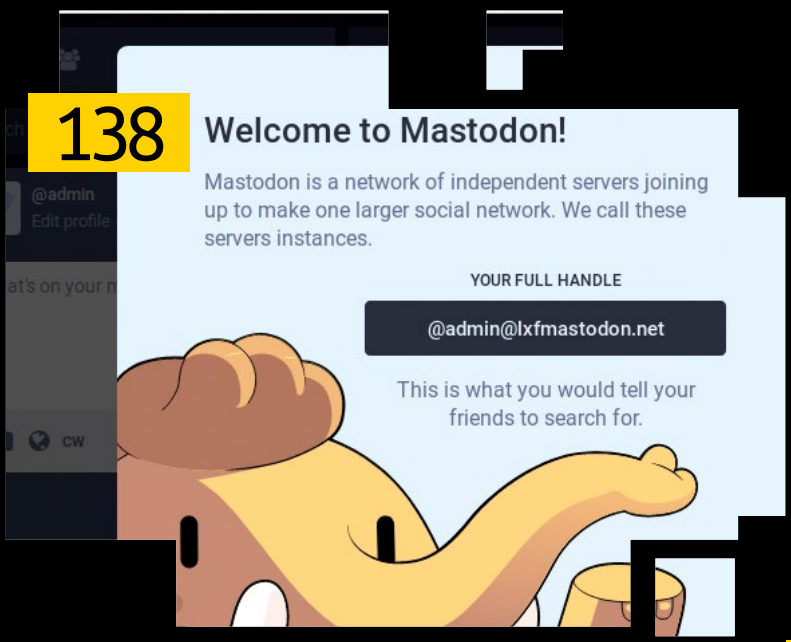


Do more



CREDIT: Marcmccomb, CC BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Line_follower.jpg

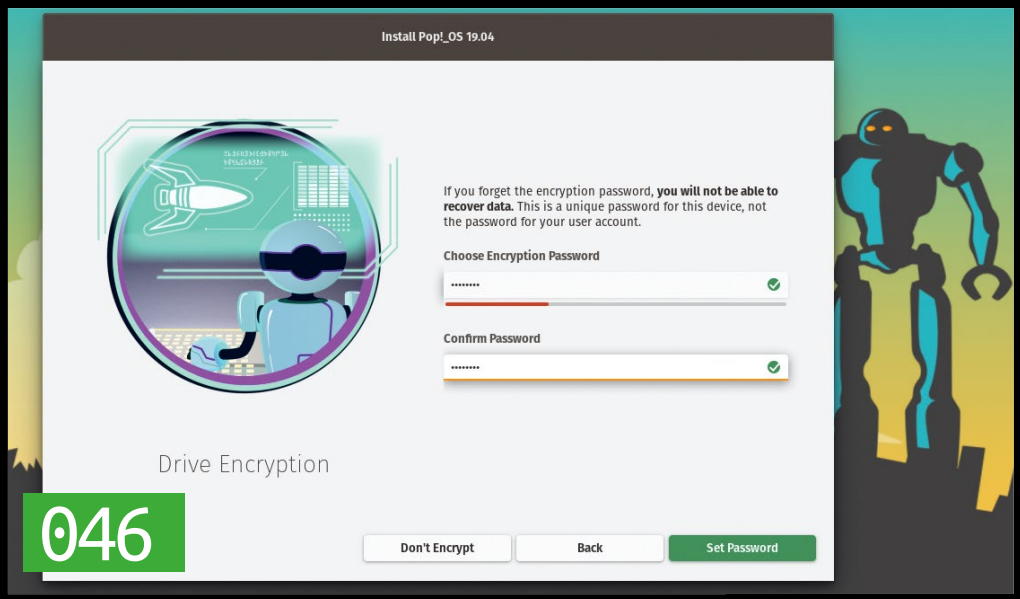
104



FOSS

Distros

010



046

NEXT-GEN DISTROS

Step right up – LXF's greatest showman Jonni Bidwell is here to unveil the finest Linux distros around!

Cast your mind back 15 years, to that pre-credit crunch optimism of the mid-2000s. Windows users were appalled by Windows Vista, and this new Ubuntu operating system was claiming that it could displace Windows. This, despite being based on Linux (something you needed to have either contributed to yourself or subscribed to an arcane journal such as *Linux Format* to understand) and thinking that orangey-brown hue was a good desktop colour.

Well, Windows may not have been wiped out by Ubuntu. But Linux has developed in leaps and bounds. It's more useable than ever and key industry players take it much more seriously now. Thanks to Valve and Vulkan, we can play thousands of Windows-only titles on our Linux boxes. More companies than ever are shipping Linux on consumer hardware. Your wireless hardware probably works with it.

Best of all, there's a fine range of distros to choose from. Ubuntu has always been a great place to start, but it's not to everyone's taste. Here, we present our pick of the next generation of distros, sure to ruffle a few feathers. Since making Linux easy is very difficult, we've got a section on useable distros, including our long-time favourite for beginners: Linux Mint.

Next, we'll explore some of the best-looking distros out there that will give you a truly modern desktop. In particular elementary OS, through its Pantheon desktop, is doing the unthinkable by making Linux simple, powerful, and – dare we say – at least a little bit Mac OS-like.

Finally, we'll look at distros that are leveraging the latest in open source technologies. You might not want these technologies (or so it seems by the number of complaints we get about Wayland, Systemd et al.). Or you might not be able to use your favourite software with them easily, but these will shape desktop Linux in the years to come.

Linux milestones

Many battles were fought – and coffee urns emptied – to get distros to where they are today. Here's a quick recap...

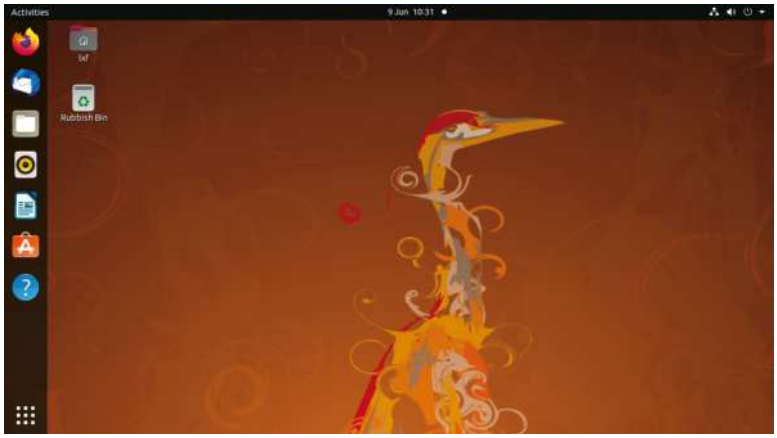
Linux distributions have come a long way since the good old days. In the beginning, of course, there were no distros (*actually in the beginning there was no Linux, and no space-time, then there was a big bang... – Ed*). You'd start with the kernel, somehow bootstrap a barebones system, fetch some GNU tools, mess with the make files, compile those packages, install them, realise you'd got your Makefile wrong, tidy up the mess. Rinse, lather, repeat. It was great fun.

More often than not you had to get these things on CD or even floppy disk in the post, unless you had access to the internet (or a friend in a computer science department). Then in 1992 came SLS, which inspired Slackware and later frustrated Ian Murdock into creating Debian. Yggdrasil, the first Linux live CD, was launched shortly after SLS, which required a gluttonous 8MB of memory and a gargantuan 100MB of disk space. The first stable version of Debian didn't appear until 1996, by which time Red Hat Linux was on the scene and all of a sudden people realised there was money to be made with that thar Linux.

It's easy to overlook the contributions of those pioneer distributions, and other giants such as SUSE and Mandrake. And indeed those of lesser-heard ones such as Conectiva (a distant ancestor of Mageia that popularised Linux in South America). There's a tendency to just focus on Ubuntu as the great humaniser of Linux.

By the same token, there's a tendency to dismiss desktop Ubuntu today as a sideshow to Canonical's commercial success. Ubuntu continues to do great things for Linux, and is an excellent distribution for beginners and professionals alike. But what happened in the mid-2000s was pretty exciting. Suddenly, here was Linux that anyone could use. It did everything that Windows XP did (except maybe talk to your wireless device). There was an office suite that could mostly open *Word* documents. Finally, something was standing up to the Microsoft juggernaut...

Things are different now. Microsoft's attitude towards open source has changed and desktop computing isn't the be-all and end-all it once was. In our **LXF262** interview with previous *Linux Format* editors, they point out there's no longer the threat to Linux that there used to be. The web is truly OS-agnostic, so there's no danger of say, your bank not supporting you if you use Linux. We take for granted the ability to watch Netflix or play games, but this would have been unimaginable just a few years ago. Today Intel, Oracle and even Microsoft (sort of) now have their own Linux offerings. A few stand-out distros have emerged that have brought genuine innovation, whether technical or ideological, to the Linux world. So let's have a look at some of them...



The Hardy Heron wallpaper (from 2008) was remastered for Ubuntu 20.04.

CHANGING TIMES

Today Intel, Oracle and even Microsoft (sort of) now have their own Linux offerings

» SOLVING PROBLEMS

Don't worry if you think something's not right with a particular Linux distribution. You're entitled to a full refund after all. Only kidding. In 2012 even Linus himself raged at OpenSUSE developers when he discovered that adding a printer (on his daughter's laptop) required the root password. Sometimes there are reasons why things are that way (in 2012 there wasn't the notion of a privileged local user; it's handled by systemd these days), sometimes it's a genuine bug, and sometimes you're just using it wrong.

These days Linus is generally calmer (except perhaps on the subject of L1D cache flushing), and his recently purchased development machine (a Threadripper 3970X which you can read all about on ZDNet at **www.zdnet.com/article/look-whats-inside-linus-torvalds-latest-linux-development-pc**) runs Fedora 32. And we'll talk about that later on. We'll also see how elementary OS is tackling unnecessary password requests over the page.

There are plenty of ways to get help with Linux, and thanks to its popularity it's highly likely someone has encountered your issue before you did. So your first act should be to always be to Google (*or DuckDuckGo if you prefer–Ed*) your problem. Beyond that, head to your distro's forums and search there. If you still feel like you've hit a new issue then describe it as best you can, and check the forum's guidelines about how to get and attach appropriate log files and hardware informations. Help the community help you.

User-friendly usability

Some Linux distributions go out of their way to make new users feel comfortable. We shine a spotlight on those welcoming distros.

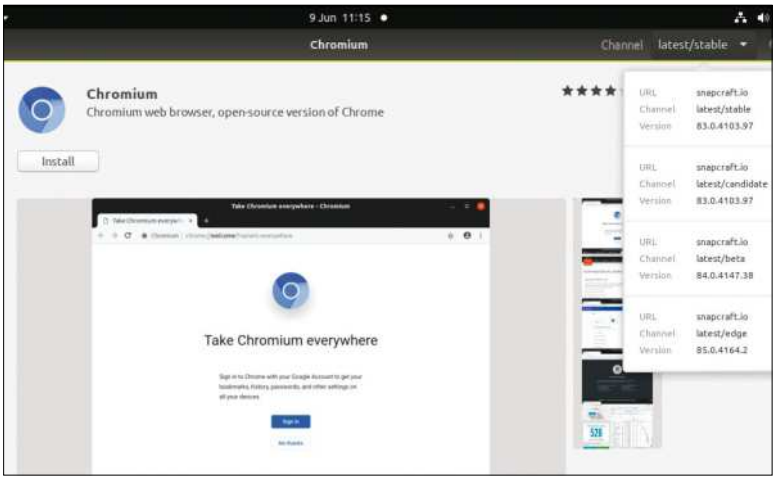
Let's look at Linux Mint first, which continues to be a favourite of ours. In particular, it's one that we still recommend to users who are taking their first steps with Linux. Initially (the 1.0 release in 2006 was a beta based on Kubuntu), it took the Ubuntu codebase and bundled flash and Wi-Fi firmware to make for a better out-of-the-box experience. It experimented with its own codebase for a couple of years, but then returned to Ubuntu's and since then the two have always been package-compatible. It enables Mint to piggy-back off the treasure trove of packages in the Ubuntu repositories, while still providing its own experience.

When Ubuntu switched to its Unity desktop, Mint offered something more conventional in the form of its Cinnamon desktop (which first appeared in Mint 13). Cinnamon was initially based on Gnome 3, but soon became its own thing. For die-hard traditionalists, they also offered the then-fledgling MATE desktop a fork of Gnome 2. Both Cinnamon and Mate continue to thrive, while Unity has been abandoned by Canonical (though the latest version lives on in the UBPorts mobile OS, and the penultimate edition is still tended to by a small but loyal community). When Ubuntu 18.04 was released, sans 32-bit ISO, Mint went ahead and produced its own (32-bit packages were still built for 18.04, this is not the case for the latest edition, so Mint 20 is 64-bit only).

We're still waiting for Mint 20 to arrive and Clem and the team have remained largely schtum about what to expect. In previous releases, it's been as much about what they've removed from Ubuntu as what they've added. For example, the 'anonymous' telemetry information that you have to opt out of sending in Ubuntu is banished from Mint.

As Ubuntu moved towards the Snap packaging format, some Mint users voiced concern about whether these would become the norm in Mint, too. They (unlike Flatpaks and AppImages) rely on a proprietary app store after all, which makes Canonical a sort of kingpin in the software distribution ring. The team reassured users that Mint wouldn't contribute to this Snap monopoly. Last year they said they'd support Snaps as long as they were useful, and "didn't become the de facto standard [for packages]". Now they've put their money where their mouth is. Try installing *Chromium* with *apt* in Ubuntu 20.04 and you'll find you get a dummy packages that installs the Snap version. This is irrefutable evidence of a Snap usurping a traditional package, and something the Mint team won't stand for.

Chromium is only available as a Snap in Ubuntu. If you want it in Mint 20 then you'll have to fetch it manually.



Mint 20 beta insights

As we write this, the first details of what to expect in the Mint 20 beta are starting to emerge. And one of the first is that no Snaps will be installed by default. The *snapd* daemon will also be absent, but you can add it if you wish. As a precaution, traditional *apt* packages will be blocked from installing *snapd*, to prevent them acting, like *Chromium's*, duplicitously.

Linux Mint made waves by giving users what they want. It sounds harsh and there are all kinds of complexities behind this, but other distros seemed reluctant to listen to their users. Alas, the most common gripe "why can't it just work?" is alive and well today (though this applies to other OSes and in fact all forms of technology). A sad trend that Ubuntu's accessibility inadvertently started was frankly unqualified users taking to bug trackers and venting their rage. People would describe their problems (badly) in angry block capitals, they wouldn't supply log files or hardware information, they wouldn't attempt to pin down the problem, they would fail to do even the most cursory search for other people having similar problems or, worse, they would jump on to similar bug reports crying "me too!" – even though their issue was different.

Developers shied away from engaging with such reports. The canned response thanking these newcomers for their submission and referring to reporting guidelines has become ubiquitous. One place Linux Mint excelled was through its community helping each other, and developers engaging with that community rather than being bothered by them. Their "Newbie Questions" forum still bears the tagline "all gurus once were newbies" and is still for the most part free of more experienced users being condescending to newcomers' questions. Even where those newcomers didn't read their helpful guide on how to ask for help.

Endlessly useable

Over the past decade some Linux distributions have appeared that go out of their way to be more useable (and less breakable) by regular humans. One such is EndlessOS. Originally tied to Endless Computers'

hardware (budget small form-factor PCs aimed at developing countries), the Debian based distro is now available to all and comes in a few forms. The full desktop edition (available in seven languages) is a whopping 16GB in size, but includes the full text of Wikipedia, several Khan academy lectures and Endless' own educational tools. It's aimed at offline, and even off-grid usage. There's a more conventional desktop version (only 3GB), a VirtualBox image and finally an image for the Raspberry Pi 4. Besides their hardware (including the Spark kids' laptop) their educational contributions (such as their immersive Terminal Two coding tutorials, see **https://terminaltwo.com**), Endless has taken a particularly innovative approach to system updates.

Such updates are a necessary inconvenience, but Microsoft has unfortunately turned a lot of people against them. Most egregiously when it force-updated thousands of Windows 7 machines to Windows 10, in many cases breaking them. But also, unless you tell Windows 10 to do otherwise, it has a habit of forcefully installing large updates at shutdown time (no good if your computer is in the same room you sleep in) or at boot time (no good if you actually wanted to use your computer when you turned it on). Another sad situation is when disk space is low, large updates will download and not be able to install. In some cases (particularly cheap tablets) this can make them entirely unbootable.

Linux updates tend not to be so disruptive, but they do on occasion break things. This is particularly upsetting for new users, especially if they're confronted with a confusing error message and a busybox shell or worse, a blank screen. It's easy to blame these issues on user error, or the Nvidia driver, but people shouldn't have to learn all about chrooting and modifying system files just to get their desktops back.

Linux Mint conceded this, and that's why it launched its *Timeshift* tool for rolling back updates. Endless OS goes one step further: The whole filesystem is updated at once. This so-called atomic upgrade uses OSTree, itself described as "Git for filesystems" (or in this case operating systems). Of course, this doesn't mean every Endless OS device ends up with exactly the same filesystem (that wouldn't work unless every device was identical, hardware and software-wise). Rather, configuration files (say, everything in **/boot** and **/etc**) are layered atop the base OSTree image, and individual applications are installed as Flatpaks. You can roll back the entire OS to a previous version of the OS (just like you can switch Git branches) at the press of a button.

While the underlying OS is based on Debian, you can't readily use *apt* and **.deb** packages (or any traditional packaging format) on Endless OS . The OSTree/Flatpak combination makes for a bulletproof platform because the root filesystem is read only. Even if the power goes out mid-upgrade, the previous incarnation of the OS will still be accessible and the upgrade can be resumed. Installing an application won't break the base image because Flatpaks are self-contained. OSTree images are incremental too (only changes between versions are stored) and so the ability to roll back to a previous version doesn't come with a huge disk space cost.

Learn about history, science and, er, celebrities with Endless OS. The possibilities are, well, endless.

Mint gives users what they want: a traditional menu in the corner, a powerful file manager and beautiful desktop backgrounds.

» NEXT-GEN APP DISTRIBUTION

Flatpaks, Snaps, AppImages: they all seek to make it easier for developers to ship software in a distribution-agnostic manner. Whereas Snaps rely on a central app store, anyone can set up their own Flatpak remote. The Flathub repo (**https://flathub.org**) is a good place to find popular apps. Flatpaks are OSTree images too, which is why the two integrate so nicely in Endless OS. When you download your first (non-trivial) application on Endless OS (or your first Flatpak on another OS), you'll find the download is very large. This is because it depends on a large runtime (e.g. Gnome) Flatpak. Once you have this you won't need to download it again for Flatpaks, which depend on it. These runtime Flatpaks are updated incrementally, so subsequent large downloads should be a rarity and (unlike deb packages) there are no issues with requiring a particular version.

Fedora's Silverblue desktop distribution brings atomic upgrades to the Fedora ecosystem. Here, a hybrid packaging system, RPM-OSTree, is used so that the base OS image can be modified in layers. This is useful for installing low-level packages such as drivers and shells. Conventional desktop programs are installed this way, and there's also support for containerised workloads.

Flatpaks and snaps also take the burden of packaging away from distros.

The new generation

Check out these wonderfully crafted distros that give the operating system from a certain fruit-named company a run for its money.

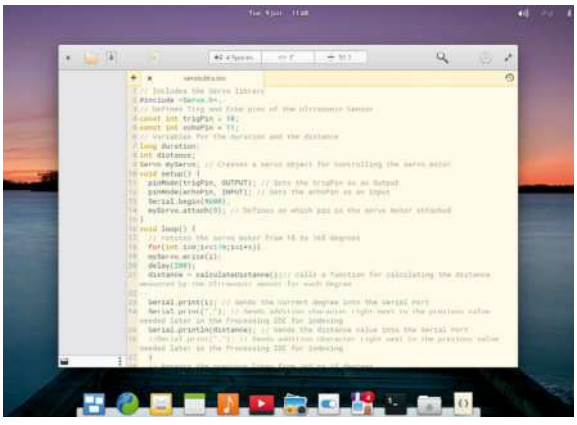
Ubuntu-derivatives are a dime a dozen. Some you could probably recreate yourself by installing a desktop and changing some default applications. But some are simply outstanding, taking the rock solid Ubuntu base and adding unique and powerful features atop it.

One such is elementary OS which, through its own Pantheon desktop, brings a sleekness and simplicity that rivals Mac OS. Instead of forcing users to read manuals (which in many cases just raises more questions), one of elementary's aims is to have minimal documentation. It aims to be sufficiently intuitive that users should be able to figure things out for themselves. Another of its design aims is "concision", keeping things simple and avoiding bloat. Finally, it strives for "accessible configuration": not bombarding the user with options, or asking for information that could be obtained automatically. This all might sound like fluffy design speak, but when you see how the app indicators convey with subtlety useful information, how the infobars gently notify you of exigent situations and how everything integrates so nicely (like a new-age type wittering on about how "it's all one, brah"), then you'll see there's substance to it all.

Contributing to elementary OS

Elementary OS is not free as in beer (*beer's free now?–Ed*). It uses a pay-what-you-want model and when you download it you're asked to make a contribution. Ubuntu and some others use this model too, and nothing bad happens if you select $0. There's nothing inherently wrong with charging money for software – even free (as in speech) software. It's even expressly permitted by the GPL. Elementary, in asking for donations rather than slapping on a price tag, is striking a balance. It's ensuring that people get reimbursed for their work, while at the same time making it available to all. If you disagree with that stance then you're under no obligation to use it. If you want to contribute in another way, you can help fix bugs or help users on the forum (as you can for any software). There's even a bug bounty program, so if you're willing to pay for a feature (or would like to get paid for fixing something), then subject to team approval and someone coding it, you can.

Fiscal models aside, elementary OS is beautiful. The Plank dock gives easy access to commonly used applications, and the bundled ones all follow a uniform design. They're all coded in Vala, which makes them integrate easily with GTK (or elementary's custom superset of it, dubbed Granite). They're free of



Code, elementary OS's text editor, is beautifully crafted. We can't guarantee our Arduino code is likewise.

» BEHOLD, THE GORGEOUS POP!_OS

It would be remiss of us not to mention, in a spread about beautiful distros, Pop!_OS. We were thoroughly impressed by the 20.04 release back in April (see our glowing review in **LXF264**). Pop (we like most everything about it, bar the awkward punctuation) is made by Colorado-based Linux system manufacturer System76. Being an OEM, they wanted to be able to ship an OS guaranteed to work with their hardware. But also, just like the other distros named in these pages, an OS that makes it easy for users to do whatever they want to do. Pop's desktop is much closer to Gnome 3 than what Mint, elementary and Solus offer, but don't be afraid. Don't also be afraid that the latest release gives you the option to experience (in a very gentle way) the joys of a tiling window manager.

Pop!_OS ships two editions: one featuring the Nvidia proprietary graphics driver and one featuring open source drivers (for AMD and Intel hardware). You'll find the latter on our virtual DVD, or you could save yourself some bytes and download it directly. System76 have contributed an awful lot to Linux, including a Firmware Manager front-end for updating firmware via *Fwupd*. In Pop, this is integrated into the update tab in the Gnome Settings application. Naturally, this works with not just System76 hardware, but anything supported by *Fwupd* (which is a lot of hardware). Pop also has a bunch of tweaks for common gripes like multiple displays, and hybrid graphics.

Pop comes with fine backgrounds and this button opens the portal to tiling utopia.

unnecessary configuration options, proving that sane defaults are possible. Most interestingly, many of them don't have an explicit Save option. They'll save as you go and reopen to the same state they were left. The AppCenter (sic) includes a section of curated programs (that adhere to elementary's design philosophy and have been reviewed by the team) as well as everything you'd find in Ubuntu's erstwhile *Software Centre* (now the Snap Store). Flatpaks are supported out of the box too via a custom tool, *Sideload*. Unlike Apple's walled-garden approach, elementary makes it possible for adventurous users to stray from the curated base camp and brave the open source jungle.

Besides looking pretty and being a pleasure to use, elementary OS is conscious about privacy. *Onboarding*, which debuted in the 5.1 release, offers to turn on location services or automatically delete temporary files (and optionally Trashed items). This launches on first run for each user (the new user selection screen is also lovely, by the way) and also guides users to the documentation, Night Light settings and AppCenter. Many users will have long-abandoned email clients, opting instead for the web-based offerings from their provider. But elementary's *Mail* program might just spur those users to make their inboxes local again.

We're looking forward to elementary OS 6.0, which will be based on Ubuntu 20.04, but the team snuck out a point release, 5.1.5, in June which brought yet more nice touches. Being asked for a password occasionally might not seem particularly onerous, but when you encounter the password prompt on a daily basis it gets tiring. Elementary aims to tackle this "authentication fatigue" through contextual authentication, or only asking for a password where it makes sense to do so. If your laptop (which typically only one person uses) asks for a password when you log in, then it shouldn't really need to ask you again when you apply updates, or even install new software. Likewise, you shouldn't be prompted for a password, for example if you just look at (but don't touch) firewall settings. You can follow all the contextual authentication work on Github at **https://github.com/orgs/elementary/projects/74**.

Parental relations

Desktop distros these days tend to all have a parent distribution. That might be something still going (Ubuntu, Debian) or something now defunct (such as Mandrake, where OpenMandriva and Mageia's roots lie). Such distros may be forks (have evolved their own codebase), or they may retain compatibility with their parents. Ubuntu is technically a fork of Debian unstable, whereas Mint is an Ubuntu-derivative. Either way it's rare that something totally freestanding comes out. Which is why we must give credit to the Solus project.

Solus started life as Ikey Doherty's Evolve OS back in 2013. The project's package manager, *eopkg*, was a fork of PiSi from Turkish distro Pardus Linux (*see reviews* **LXF259**). At the time (Evolve remained in alpha until 2015), most interest was centred around its unique Budgie desktop, which used Gnome 3's underpinnings to create a stylish yet traditional experience. Due to a trademark conflict (with the then-UK secretary of state no less) Evolve was renamed Solus, and Solus 1.0 was released at the end of 2015. By then Budgie had acquired its characteristic Raven sidebar (nevermore dig deep to tweak audio or see appointments with Lenore) as well as a substantial following.

We had trouble deciding which of Solus's wonderful backgrounds to capture alongside the Raven sidebar, but cat pictures win every time.

By 2017 Doherty announced that he would be working full-time on the project, and Solus had moved to a rolling release model. The team had developed their own tool, *ferryd*, for rapidly deploying changes to their repositories. While not a strictly Solus project, work continued on his Linux Steam Integration tool (which forced Steam to use distro libraries rather than the antiquities found in its own runtime). Solus's driver management tool saw its first release and Ubuntu Budgie (the desktop has long been available outside of Solus) became an official Ubuntu flavour. Ikey left the project in 2018, leaving it the capable hands of his co contributors, and the rest as they say, is history. Budgie uses the semi-classical Brisk menu, designed in conjunction with Ubuntu MATE, and (like elementary's Pantheon) all development is done in Vala.

SUPPORTING YOUR DISTRO

Elementary OS is not free (as in beer). It uses a pay-what-you-want model and when you download the ISO you're asked to make a contribution

The current Solus release, 4.1, comes in four editions (Budgie, Gnome, MATE and Plasma) and packs a range of multimedia features. For gamers, tweaks to the default file limits have been raised so that *ESync* can be used for better performance in *Wine* (or *Proton*, via Steam).

Taking software solace

Since Solus maintains its own repos, you won't find as many packages as in, say, Ubuntu's traditional **.deb** repositories. However, Snap is installed out of the box, so you can now avail yourself of the many offerings in the Snapcraft store. Flatpak and the Flathub remote are only two lines away too:

```
$ sudo eopkg install flatpak xdg-desktop-portal-gtk
$ flatpak remote-add --if-not-exists flathub
```

Eventually these will be integrated in a new plugin-based *Software Centre*, and *eopkg* will be replaced by a next-generation package manager dubbed *Sol*. We look forward to the next Solus release, and indeed Budgie 11.

Next-gen distro tech

Shiny and easy-to-use distributions are one thing, but which distros ship the latest Linux tech? Keep reading to find out...

Fedora is Red Hat's community distro, where new technologies are honed before they make it into their business, meaning Red Hat Enterprise Linux (RHEL). Fedora was the first distribution to see *systemd*, *Pulseaudio* and SELinux (the NSA-contributed access control mechanism). Those titles might send shivers down the spines of some readers, but like them or not they're here to stay. It's historically been the best way to experience unadulterated Gnome 3 (stop shuddering you lot). That desktop offers the best Wayland experience, and Fedora was the first to offer Wayland by default (still with the shuddering?). So it's nothing if not a trailblazer distro.

The latest Fedora 32 even enables *Firefox's* new Wayland backend. And if you're feeling adventurous, follow the guide in our Web Browsers feature (*page 42*)

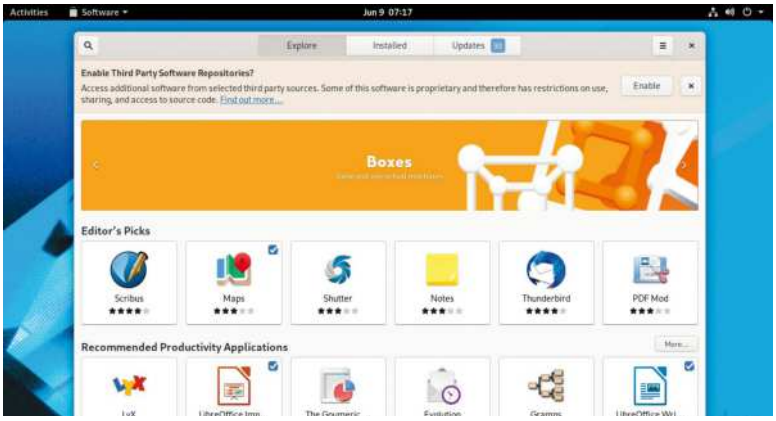
LINUX HITS THE MAINSTREAM

With laptop offerings from both Dell and Lenovo, we can direct users who want a rock-solid Linux offering out of the box there with confidence

to enable Webrender and VA-API acceleration (which you can in turn read more about on Martin Stransky's blog at **https://mastransky.wordpress.com/2020/06/03/firefox-on-fedora-finally-gets-va-api-on-wayland**). Indeed VA-API acceleration, which appeared in *Chromium,* debuted in Fedora's build over a year ago. So if Fedora receives new features, smoother browsers, why on earth aren't you using it?

Occasionally Fedora is sidestepped because of its philosophy on software freedom. If you want proprietary software (including the Nvidia driver), or patent-encumbered multimedia codecs, these have to be added through a third-party repo such as RPMFusion or Negativo 17. The codecs issue is much less of a big deal nowadays. Popular streaming sites all use open formats (or at least, in the case of h.264, ones where an open source codec is available) and MP3 became a patent free format in 2017. A free AAC codec was made available shortly afterward, so you can play all your ripped CDs from yesteryear out of the box. Or you could rip them again in a lossless, free format such as FLAC.

Third-party repos are only a click away in Fedora.



Make it easy on yourself

Setting up a traditional third-party Fedora repo requires a certain amount of effort. More so than setting up a PPA for Ubuntu, which is perhaps why it used to be a bit harder to find certain niche software for Fedora. Bear in mind that PPA overload was definitely responsible for the demise of many an Ubuntu install, and third-party builds should always be treated with caution. Nowadays, a new technology Copr, takes all the hassle out of building unofficial Fedora packages. Fedora even offers a Copr build service that will provide you with your own Dnf/Yum repository. Just upload a source RPM file, select your build targets, and you're done. Read more at **https://developer.fedoraproject.org/deployment/copr/about.html**. And, of course, more and more developers are turning to Flatpaks and Snaps (although the latter is disabled in Fedora's build of *Gnome Software*). So Fedora's comparatively small, but carefully curated repos are no reason to shun it.

Fedora made the news recently with the announcement that Lenovo will be shipping laptops powered by Fedora Workstation. Writing on his blog Christian Schaller, senior manager for desktop at Red Hat, says "Our engineering team here at Red Hat has also been hard at work ensuring we can support these models very well, be that by bugfixes to kernel drivers or by polishing up things like the Linux fingerprint support". We often have queries about Linux on laptops from readers, and without a particular model to hand it's difficult to give solid advice on compatibility. Now, with laptop offerings from both Dell and Lenovo, we can direct users who want a rock-solid Linux offering out of the box there with confidence.

Christian has been adamant about "draining the swamp" over the course of his past six years at Red Hat. He's not talking about rogue politicians, either. Rather he's addressing desktop issues that for too long have been hastily patched over, where ground-up rewrites were required. He original credits the phrase in this context to a talk by Jim Gettys, whose thoughts you can read at **https://mail.gnome.org/archives/foundation-list/2002-May/msg00005.html**. The latest Fedora includes Gnome 3.36 (the same as Ubuntu 20.04) and

we're impressed with the distinct lack of swamps. You may have run into *PulseAudio* issues in the past, or you may not see any reason to use Wayland over X11, but these technologies were invented to solve problems. And solve them they do. You might also be interested that *PipeWire*, Wim Tayman's offering to unify media handling, is included by default in Fedora. *PipeWire* (*see features* **LXF232**) can now do pretty much everything *Jack* (the rather complicated interface for pro-audio set ups) can, and the next step is to have it take on the duties of *PulseAudio* too.

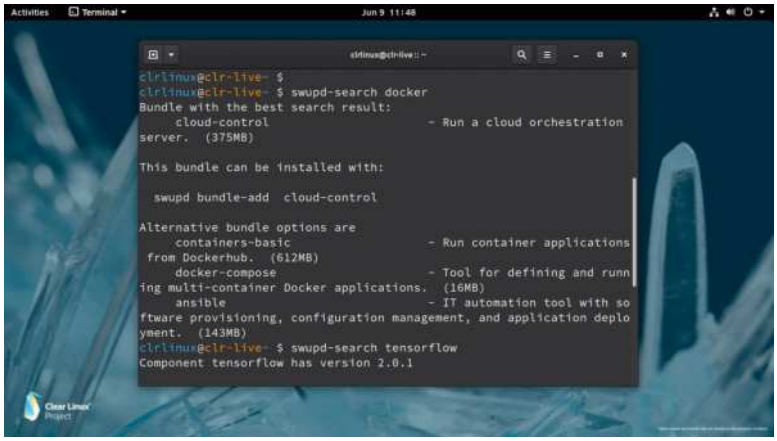
If you've ever run Linux on a machine with not enough memory, you'll know that things can get very unpleasant very quickly. Apart from the slowness of using spinning-rust based swap space (which is not an option on some systems), you might find yourself at the hands of the OOM (out of memory) killer. Applications will be slain more or less arbitrarily. Even worse though, is when OOM can't react quickly enough and your desktop grinds to a halt. You can't move the mouse pointer, you can't switch to a virtual terminal, you probably can't even SSH in to kill things manually. We've run into this memory-pressure situation a few times when running one too many virtual machines. Like when you're writing a big feature about lots of distros and then it makes it late (*Are you saying the OOM-killer ate your homework? – Ed*). Anyhoo... Fedora 32 includes the *EarlyOOM* daemon, which should step in before it's too late.

Fedora goes much further than the desktop. Naturally it has a server offering, and we've already mentioned its Flatpak- and container-focused Silverblue desktop effort. But since Red Hat purchased CoreOS in 2018, it's now in charge of the distro formerly known as CoreOS Container Linux, now dubbed Fedora CoreOS. This will very likely be the future for managing container-based workflows securely, and in particular massive server deployments. And from the very big to the very small, in the form of Fedora IoT, its offering for the tiny things that will be running our cities and lives in the coming years.

A Clear winner

One distro we haven't covered much, but perhaps should, is Intel's Clear Linux. This is aimed at professionals using Intel hardware to do advanced things with container and cloud technologies. In its own words it's "not intended to be a general-purpose Linux distribution". Be that as it may, it consistently scores favourably in Phoronix's tireless benchmark comparisons and a desktop edition is available so you may want to have a gander. Clear Linux abstracts application packages into the concept of Bundles, which enable a bunch of related programs to be installed at once. Since Clear Linux has its own repository you might not find your favourite applications, but this is exactly part of the problem Flatpaks aim to solve, and installing the Flatpak bundle is easy.

Clear Linux bundles use compiler optimisations, enabling AVX512 vector instructions on suitable endowed hardware. So, for example, if you're running TensorFlow, you'll benefit from kernel tuning, AVX routines in Glibc, AVX optimisations in Python, tweaked Numpy/Pandas modules and finally optimisation at the very top of the stack in Tensorflow's Eigen component.



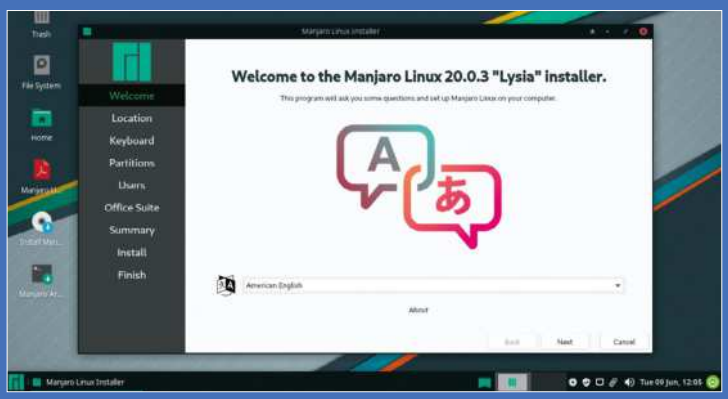
Clear Linux's swupd tool will get you bundles and have you controlling clouds in no time. There are some pleasing background on offer, too.

Clear Linux uses a stateless design concept so the whole OS can be effectively factory reset by clearing **/etc** and **/var**. This doesn't work on other distributions so please don't try it. But do try all of the distros we've mentioned here, and feel free to shout at/email us for not mentioning your favourite next-generation offering. LXF

» WHAT ABOUT ARCH?

Arch users may be shocked to find their distro of choice not mentioned in this page on new technology, and perhaps they have a point. But then so too might Gentoo or Linux from Scratch users, who also have access to the latest versions of everything, and can also compile it with whatever optimisations suit their CPU. We've focused instead on what distros offer out of the box, and with Arch, and more so with Gentoo, well, you barely get a box. Be that as it may, Jonni uses Arch, and so should you.

Manjaro did the unthinkable and made Arch Linux easy to use, and continues to be one of the most popular desktop distros around. Manjaro doesn't (directly) use the same repositories as its progenitor, since things move quickly on Arch, and Manjaro has a release cadence to follow (and a desire not to break its own packages). So you won't find quite the same bleeding-edge versions as you would in Arch, but the selection will be far from stale. And bugs that may have found their way into the stable Arch repos will have been ironed out by the time they reach Manjaro. We like that Manjaro is getting the Xfce desktop some attention. Lightweight doesn't have to mean light on features. And if you want a beautiful take on KDE's Plasma desktop you can also get that through Manjaro's Plasma edition, which we snuck onto our virtual DVD last month.



Taming Arch Linux into something friendly? Well, now we really have seen everything!

Roundup

Antergos » ArchLabs » KaOS »
Manjaro » Netrunner Rolling

Jonni Bidwell required judicious application of Jura whisky to fend off the germs that tried to interfere with this *Roundup*.

Arch-based distros

Ever more distros are being built upon Arch Linux. Once **Jonni Bidwell** would have called this sacrilege, but now he finds it a glorious development.

HOW WE TESTED...

We installed each distro on a Dell XPS13, which enabled us to test HiDPI and touchscreen support. We checked that the distros happily installed alongside other operating systems without breaking them. Further testing was done in virtual machines (limited to 3GB of RAM), which gave us some idea of how the distros would run on older hardware.

We looked at what came with an out-of-the-box install, and whether this struck the right balance between usefulness and bloat. Since we've focused on distros providing a desktop, we looked at how stylish and effective those desktops were. Bear in mind that it's possible to install and configure alternative desktops to your liking.

We compared repositories to Arch Linux's, renowned for offering the very latest software and making minimal changes to upstream releases. In particular, we looked at the desktop packages used by each distribution. We also checked how each distro integrated with the Arch User Repository (AUR).

Few would disagree that Arch is a fine distribution: cutting-edge, customisable and possessed of an enthusiastic community and excellent documentation. But it's not for everyone.

Many people are put off by the lack of an installer: having to manually partition disks and `chroot` into a bootstrapped install to get things going doesn't really score any user-friendliness points. Those that surmount that obstacle are met with a new challenge: namely, building the minimal install into something useable. If you're installing it for desktop use, you'll find getting that desktop to look and play nice is decidedly non-trivial. In short, it really makes you appreciate the efforts that other distros go to, to have all this set up out of the box.

But the goodness of Arch can be married with the convenience of other distros. Imagine never having to add a PPA to obtain new releases. Imagine being able to rebuild custom versions of stock packages. Imagine, and then imagine no more. We're focusing on desktop varieties here, so we'll be paying particular attention to modern desktop innovations, but people have used Arch Linux as the base for all kinds of things (security, Raspberry Pi, 32-bit hardware, the list goes on), such is its flexibility.

Install and first run

Does the installer welcome you?

Antergos impresses right from the outset with the stylish *Cnchi* installer. You can choose to set up popular applications and desktops. These all need to be downloaded, which puts the kybosh to offline installs, but it does do installees the courtesy of offering to find the fastest mirrors. If you're worried about the install going south and having to start over then you can opt to create a recovery partition on the target device. This is a nice touch, but mercifully not one we had reason to use. You can even choose between GRUB2 and systemd-boot, encrypted installs and even using ZFS.

Manjaro's live medium starts with a slightly spartan menu for selecting keymaps, free vs non-free drivers or adding kernel parameters. Once the *Calamares* (*see* **LXF233**) installer fires up we're met with the usual options (LVM and encryption) and you can choose between wiping your drive, installing alongside other OSes, or manual partitioning. Once you boot into Manjaro, a welcome screen displays release information and provides links to the Wiki. There's a handy tool for installing popular apps, too.

Netrunner, being a Manjaro descendent, has a similar installation process. Newcomers might be a little upset to find the Readme icon on the desktop takes them to a 404 page, with a mixed content warning, but this is a cosmetic thing. KaOS also uses the *Calamares* installer. XFS is a slightly left-field choice for a default filesystem, but why not? Once installed the *Croeso* (Welsh for "welcome") tool greets you and provides shortcuts to documentation and common system settings. This is handy if you don't know where Plasma hides these things, and even if you do it's nice to have them all in one place. *Croeso* has a few more options than Manjaro's welcome tool.

Antergos is something of a desktop shape-shifter. Much like its Arch parent, it can be anything that you want it to be.

ABIF, Archlabs' Ncurses installer isn't newbie-friendly, but offers the same options as Antergos and carries out a full system update before rebooting. Since the live medium hasn't been updated since July 2018, this is a fairly lengthy process. Once you reboot, a menu enables you to install popular applications, fonts and so forth. If you don't like logging in from the console, the LightDM display manager is readily available from here, too.

VERDICT

ANTERGOS	**9/10**	**MANJARO**	**8/10**
ARCHLABS	**6/10**	**NETRUNNER ROLLING**	**8/10**
KAOS	**7/10**		

Antergos's Cnchi sets a new standard for Linux installers. ArchLabs's textual experience can't compete, but console warriors won't mind.

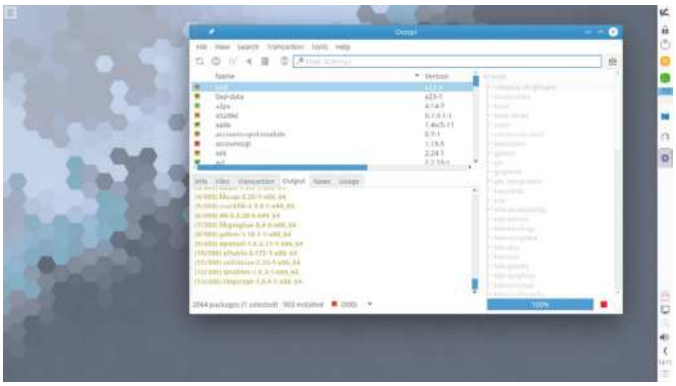
Repositories

How do these repos stay up to date?

Be mindful that we're using the stock Arch repositories as a baseline here, and short of packaging things yourself or using snaps this is pretty much the fastest way to get the latest upstream releases.

Manjaro takes a leaf out of Debian's book, offering Stable, Testing and Unstable branches. The Unstable packages track around three days behind the Arch repo, which should be fresh enough for anyone. Packages make it into stable "when they're ready", but this is usually pretty swift unless some exigency arises. At the time of writing there's a stability issue with new versions of Systemd, but otherwise we found no major differences. Netrunner Rolling is based on Manjaro's Stable channel (though a handful of custom packages come from its own repo), so it can enjoy the 12,000 odd packages in Manjaro's repos.

KaOS tends to lag behind a little further, and the only graphical apps you'll find in its repos are Qt ones, such is its KDE Plasma focus. If you need anything else, you'll have to build it yourself. You can do so from the KaOS Community Packages repo, which given KaOS's limited repos, vastly widens its scope of interest. KaOS uses Plasma 5.14, compared to Netrunner's Plasma 5.12, and both currently use version 18.08 of the KDE applications suite, hopefully the just-released 18.12 apps bundle blesses these KDE distros soon. Finally, Antergos maintains a handful of its own packages, but otherwise relies on the Arch Linux repos, and ArchLabs uses exclusively the Arch repos.



KaOS has only a couple of thousand packages in its repositories. This is a fraction of what you get with Arch, but maybe all you need.

VERDICT

ANTERGOS	**8/10**	**MANJARO**	**9/10**
ARCHLABS	**8/10**	**NETRUNNER ROLLING**	**7/10**
KAOS	**8/10**		

Manjaro's channels cater to all users, and their generous repositories are almost as well-stocked as Arch's.

Style

Which distro offers desktop panache?

Comparing different desktops is a tricky business, especially when we have species from all over the spectrum. We're not going to mark Archlabs down because it uses a lightweight window manager, and we're not giving extra points to Antergos because it uses a readily available (though undeniably cool) icon theme.

Instead, we'll focus on how each desktop is styled relative to the components installed. We're interested in icons and colour schemes working together, considered layouts, and freedom from clutter. Since we're au fait with setting up various desktop environments from scratch in Arch, we'll give credit for sorting out common quirks or deviations from the defaults (as long as they're useful deviations).

Whatever desktop you end up using, since Arch always ships the latest releases, you'll probably end up with something that looks and acts better than it does in other distros. And if you don't, well you might at least have found a brand new bug, or you might have caught a glimpse in to what users of other distros will be kvetching about in a few months.

Antergos Gnome 7/10

Antergos is our only Gnome offering, but even before we'd got there we were impressed with the stylee console font setting during boot. Dash-to-dock provides a launcher on the left (à la Unity and Budgie), emblazoned with icons styled in the bold, Numix Square theme which is used throughout the desktop. The bright orange directory icons in Files might not be to everyone's tastes, but we liked them. Tweaks and the User Themes extension are installed for tweaking and theming things to your taste.

If you choose to use Gnome with Antergos then you'll probably want to replace the clunky-looking LightDM greeter with GDM, and there's a wiki page dedicated to doing just that. If there were an award for 'largest title bars' then Antergos would win, but Gnome apps that use new-fangled GTK3 headerbars look quite comfortable with them.

ArchLabs 9/10

We like the minimal, dark stylings applied to OpenBox and the Polybar panel. Thanks to Compton, some shadows and fades mean the desktop, albeit understated, still looks modern. Archlabs' colourful icon theme contrasts nicely with all the darkness. You can switch Polybar for Tint2, or add a dock (such as Plank or Docky), all from the comfort of the Preferences menu.

If you feel Conky is polluting your desktop background, then it can be taken away too. It does provide some helpful hints though, most importantly that the Drun launcher is available by pressing the Super key. The visual cues on the desktop pager (Home, Web, Files and Pictures) are great for making the most of four virtual desktops. A few Xfce components are used and a few more – or indeed the whole of Xfce – can be added from the welcome menu.

Package management

How do our candidates tame Pacman and integrate with the AUR?

All the distros enable you to use excellent Arch's Pacman tool, but this is probably not ideal for people who want to do things graphically. The *Pamac* GUI is included in Antergos and the Xfce and Gnome flavours of Manjaro. This is a little bit slicker than the *Synaptic*-like *Octopi* used by Netrunner and KaOS (and Manjaro KDE). KaOS includes a GUI cache cleaner and package log, but command-line tools can do these tasks just as well. Netrunner also offers the appstore-style KDE *Discover*. ArchLabs enables *Pamac* to be installed from its welcome menu, but doing things CLI-stylee seems more fitting here and, speaking to that, it includes the text-based *Pacli* front-end. *Pacli* can help with downgrading packages, something not supported in Arch Linux but something that is on occasion necessary. Manjaro's kernel tool will advise you of new kernels, it's in Netrunner too, but we found we had to invoke it manually to upgrade to a new major version. This could've been impatience on our part.

Pacman can be further augmented by various helpers for building and installing custom packages from the famously useful Arch User Repository (AUR). *Pamac* and *Octopi* both show new package notifications and can optionally integrate with the AUR. However, AUR packages may not get on with anything from outside the Arch repos, and so some caution ought to be exercised on those distros that use custom repos, particularly Manjaro. Currently, ArchLabs (where you're least likely to run into AUR conflicts) uses the *Aurman* helper, but since development of this is fizzling out *Yay* is probably a better bet. Alternatively, *Pacli* can work as a front-end to the once premier, sadly now-defunct *Yaourt AUR* helper, too.

VERDICT

ANTERGOS	8/10	MANJARO	9/10
ARCHLABS	6/10	NETRUNNER ROLLING	7/10
KAOS	8/10		

Manjaro enables you to care for your packages any way you want. ArchLabs loses out for a lack of GUI offerings, but again, this may not matter to you.

KaOS 7/10

KDE has long been noted for its configurability. This can be a curse as well as a blessing. KDE 4 in particular dispersed options all over the shop, and also shipped with a near-infamous window glow enabled by default. KDE Plasma 5 has a perfectly cromulent default setup, but offers plenty of options to manipulate this.

Being a KDE Plasma affair, we were expecting KaOS to show us something a little different from usual KDE layouts, and that it certainly did. A menubar on the right-hand side of the screen, and Dolphin demonstrating similar right bias with its navigation panel. You can probably get used to this, but it's a jarring first experience. Call us old-fashioned (*'old' is more accurate – Ed*), but we were much happier with the panel at the bottom, and the file manager navigation panel on the left. We heartily approve of the Powerline terminal stylifier, though.

Manjaro 8/10

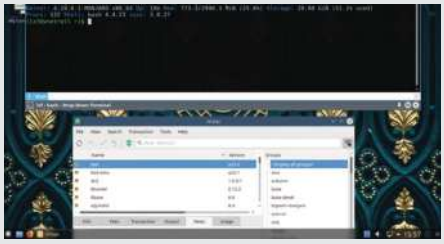
The principle Manjaro edition is the Xfce one. This isn't going to resonate with people wanting desktop bling, since Xfce hasn't changed in nearly three years. But it's functional, far from ugly and gentle on system resources. It also features display scaling, so 4K users won't be left squinting at tiny fonts, but setting this up is more involved than KDE/Gnome.

Remember that official Manjaro flavours exist for those desktops, and community flavours exist for others. Traditionalists will appreciate the inclusion of a shade control (for 'rolling' windows into their titlebars), and the ease and configurability of appearance/window manager settings. The default (Papirus-Maia) icon theme is bold without being overbearing, and there are several other icon and GTK themes to choose from. Kvantum theme manager can jazz up Qt apps, but that'll be more useful in the KDE edition.

Netrunner Rolling 7/10

Netrunner's Plasma setup features the full screen application dashboard, which like Gnome's can also search documents. The large fonts look a little odd, though. We much prefer Plasma's hybrid launcher, and you might prefer the classic cascading menu used by KaOS. The black panel looks stylish and we can't get enough of drop-down terminals so we applaud the inclusion of Yakuake.

The My Computer and Network icons on the desktop seem a bit Windows-ey, but are easily ignored/expunged. The Breeze AlphaBlack theme, powered by modern versions of Plasma (5.14) and Qt (5.11) give some neat transparency effects. Minor annoyance about the dead link on the desktop, but plenty of nice wallpapers are included. The Breeze theme has a GTK counterpart, so such applications will conform to the desktop finery. No desktop widgets are on by default. Good.



Documentation and support

Where can you turn when it all goes sideways?

The Archwiki is a truly great learning resource. It's well-written (and consistently formatted), covers pretty much everything you can imagine, and is kept up to date. A lot of what you'll find there is applicable to other distros, all the more so if Arch is their progenitor. The Arch forums are likewise a veritable fount of knowledge, but beware – fools are not suffered lightly there.

In terms of bespoke support Manjaro shines ahead here. It has its own mailing lists, bugtracker, IRC channel, a comprehensive wiki and friendly (relatively speaking) forums. There are also subforums for discussion in about 30 other languages, because Manjaro users are a diverse bunch. Netrunner benefits from this by osmosis, but they too have their own (rather more modest) forum.

The Antergos wiki appears pretty threadbare, although it does have a thorough guide to setting up dual-booting on Windows 10 which ought to appease newcomers. It also has a thriving, multilingual forum and an IRC channel. The ArchLabs forum is likewise buzzing. However, beyond some handy hints about the included programs and customising the desktop there's not a lot on the wiki. The smaller distros really benefit from community involvement and so if you solve some niggling problem (whether or not someone's asked about it) do consider adding your knowledge. For the greater good and all that.

The KaOS team use several bugtrackers and has a quiet forum. Its website has a handy guide to asking and reporting things sensibly and the world would truly be a better place if everyone read these kinds of things.

VERDICT

ANTERGOS	8/10	MANJARO	9/10
ARCHLABS	7/10	NETRUNNER ROLLING	6/10
KAOS	6/10		

Manjaro's huge userbase means a strong community, many of whom won't default to "RTFM" if you pose a genuine question.

Notable additions

Sometimes it's the little things that make all the difference...

Setting up printers in Arch is a bit of a faff, but Antergos and Manjaro include Redhat's *system-config-printer* utility. All our distros bar ArchLabs include the *HP Device Manager*, which is great if you have such devices to manage.

Netrunner includes Grub customizer, a nice touch, and like Manjaro also includes out-of-the-box support for hibernation. *KDE Discover* is a good way to, um, discover new applications and Plasma addons. It's much more solid than it used to be and we found it much less annoying than *Gnome Software*. Netrunner's Firefox install comes with uBlock Origin and several popular blocklists installed, making the web a marginally less obnoxious place.

KaOS includes a Wayland session which is a nice/brave touch for a KDE distro. Gnome on Antergos will try and start a Wayland session. Being lazy typists, we were sad to see the `bash-completion` package missing from Manjaro and Netrunner, but heartily approved of KaOS's Ash configuration and the tab completion there. Manjaro and KaOS both feature simple frontends for the UFW firewall. We think it's high time that intermediate and higher users start paying attention to which packets get in or out, so applaud both these distros. Antergos and ArchLabs both retain Arch's customisability, primarily by providing direct access to the multitudes of packages in the Arch repos. We think more people should give lightweight desktops a chance, and liked the Openbox/Polybar/Compton setup that came with ArchLabs. Setting this up from scratch is much harder than recreating KDE/Gnome/Xfce setups.

Netrunner includes the retro Burgerspace game. It's just a shame that we didn't have time to play it much. Honest.

VERDICT

ANTERGOS	**7/10**	**MANJARO**	**8/10**
ARCHLABS	**8/10**	**NETRUNNER ROLLING**	**8/10**
KAOS	**9/10**		

There are hidden gems in all our offerings, but the KaOS devs have really thought about what their users will appreciate.

Included tools

What's the out-of-the-box experience?

Antergos installed a light selection of the Gnome application suite, but more than enough to get started. Having Gnome's new *Documents* tool is a nice touch, likewise *Weather*, *Maps* and *Books*. There's no *LibreOffice*, so paper-pushers will need to do that themselves, but *Chromium* is installed which will keep 50 per cent of everyone happy. Critics of *Gnome Software* will be appreciative of its absence.

Manjaro (and by extension Netrunner Rolling) includes a hardware detection tool (*mhwd*) that runs on install and can be conjured up post-install from the settings manager or terminal. There's a handy tool for adding/removing kernels too, which is great if you run into hardware regressions. It's doubly useful in Netrunner Rolling if you need to roll with a newer (or even real-time) kernel. Less useful in both these distros is the Flash player.

Gamers will appreciate the shortcut to the Steam installer included in the Manjaro menu. And some will appreciate the links to the *MS Office Online* web tools, but others might just see these as needless menu overcrowding. NetRunner comes installed with a number of multimedia applications. There are some fun games too and we approve of the *GIMP-Inkscape-Krita* graphical trio. *Skype* and Steam and *Virtualbox* are available, too.

KaOS features the *Elisa* music player, the *Karbon* vector graphics tool and the *Seafile* syncing utility. What distinguishes it from the rest is the use of the *Calligra* office suite. This looks stylish in a Plasma setting and is in many ways less clunky than *LibreOffice*, but will be hard for some people to get used to. The QtWebEngine-powered *Falkon* (formerly *QupZilla*) is a great web browser if you don't need anything fancy, but some will be itching to get something more mainstream installed.

ArchLabs is proud of its minimalism but you can install anything you need from its welcome menu. Music and movies are taken care of with *mpv* and *Audacious*. In a nod to developers it includes the *Geany* code editor, as well as Arch's base-devel package group. We approve of the inclusion of *Nmap*, and *Firefox* will please a different 50 per cent of users.

We're not sure we know anyone who uses Handbrake, Vokoplayer and gmusicbrowser. If you do, Netrunner has you covered.

VERDICT

ANTERGOS	**7/10**	**MANJARO**	**9/10**
ARCHLABS	**6/10**	**NETRUNNER ROLLING**	**6/10**
KAOS	**7/10**		

If you think less is more, pay no heed to ArchLabs score. Manjaro have really sorted the wheat from the chaff though.

Arch-based distros
The Verdict

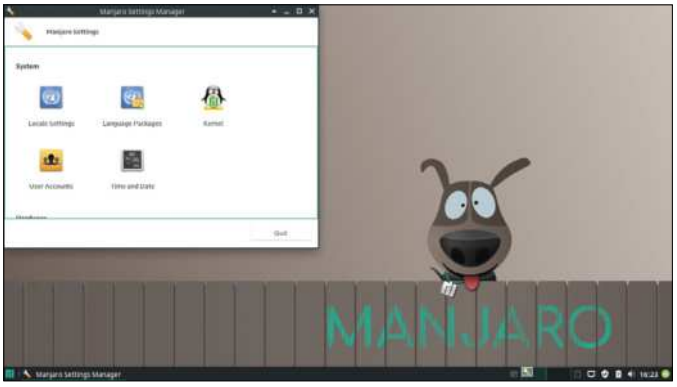
For some users, having a purely GTK or Qt system will be a boon, even if it drastically restricts the number of applications that are available to them. For those people Antergos, and in particular KaOS, may have some appeal. Others will be happy that cross-toolkit theming is in much better standing than it once was and not mind installing both toolkits. Storage is cheap, after all.

In many ways we felt that KaOS should be offering newer KDE packages, being as it does hit the Arch repos soon after release. Of course, KDE neon is considered the premier distro for showing off the latest and greatest on planet Plasma, so maybe there's no point competing with that. The Blue Systems sponsored Netrunner Rolling is a fine effort, but we felt it was trying to cover all bases.

Fantastic efforts of each distro notwithstanding, users who are unfamiliar with Arch and its ways of working will sooner or later find themselves confused. Manjaro may be the most user-friendly of this month's picks, but Ubuntu or Mint it is not, and treating it thusly will lead to disappointment. (*Life is full of disappointments, but that's a story for another Roundup.*) You'll probably have to get your hands dirty at some point. Conversely, if you're familiar with Arch then you may very well prefer to stay there (*up on that high horse you mean? – Ed*), or perhaps take inspiration from these distros and apply it to your Arch install. We were certainly enamoured by ArchLabs' desktop zen, and what can be done with a less than 200MB memory footprint.

Manjaro does a great job with its flagship Xfce edition. It shows that this desktop is capable of looking good as well as being considerate on resources. It's become hugely popular over the past year, not so much because of people's gripes with Ubuntu, but because it's a darn fine distro that gives people exactly what they want. When Xfce 4.14 is released and journalists become tired of making jokes about that, this will surely become ever the more greater.

The Gnome and KDE flavours are fine too. Great if you prefer the latest desktop fripperies, but we admire Manjaro's boldness in focusing on Xfce.



1st Manjaro 9/10

Web: https://manjaro.org Licence: Various
Version: 18.10
Manjaro has tamed Arch into something both humans and robots can enjoy.

2nd ArchLabs 9/10

Web: https://archlabslinux.com Licence: Various
Version: 2018.07
You don't need an ancient machine to enjoy a lightweight distro.

3rd Antergos 8/10

Web: https://antergos.com Licence: Various
Version: 18.11
Fantastic installer and a great all-round distro.

4th Netrunner Rolling 7/10

Web: https://netrunner.com Licence: Various
Version: 2018.08
KDE Plasma and Arch are a winning combination, but still a few rough edges.

5th KaOS 7/10

Web: https://kaosx.us Licence: Various
Version: 2018.10
One for Plasma/Qt purists, but this makes it limiting for everyone else.

» ALSO_CONSIDER

Several Arch-based distros have come and gone over the years, but some have stood the test of time and new ones spring up regularly. It's common for distros to add non-free software to improve the user experience, but Parabola and more recently Hyperbola go to the other extreme. These two, as well as being named after curves, are both FSF-approved and use the Linux-libre kernel. Hyperbola takes inspiration from Debian, using an older, tried 'n' tested kernel. Arch may have stopped supporting 32-bit, but the Arch Linux 32 distro is alive and well.

Arch-XFerience and Anarchy Linux, both Xfce affairs, are worth a look. Chakra Linux, a KDE offering, impressed us last year, but hasn't been updated since then. Those wanting something even more lightweight than ArchLabs should check out ArchBang, which was inspired by the now-defunct CrunchBang. In a roundabout way this inspired ArchLabs too.

USING BSD FOR LINUX USERS

Aaron Peters is a stranger in a strangely familiar land as he dives into installing and using Linux's open-source cousin FreeBSD...

If you're reading this, then you have at least a passing interest in free software. Maybe it's limited to Linux, but the FOSS community is bigger than that. There's plenty of software projects that run on other operating systems, and also a variety of open source Un*x clones.

The BSD family of operating system is an important example, representing some of the first systems truly free from legacy Unix code, and can be seen as at least an uncle to Linux as we know it today. What started as one system, the Berkeley Software Distribution (BSD) went through a variety of versions (a book in itself), with different projects and vendors. Today's family of BSD systems include NetBSD, OpenBSD, and TrueOS. Of these, FreeBSD is considered the most 'general purpose' and is arguably the most popular. Notably, Apple used FreeBSD code as the basis for the Darwin project, an OS that became Mac OS X (and now macOS).

If you're looking for a desktop system to install and immediately be productive with, there's better options for you (see the sidebar for some other members of the BSD clan). But this article intends to be a learning exercise to see how this little Unix-y devil stacks up. And FreeBSD offers that learning opportunity by being more hands-on than other desktop-oriented systems, whether Linux or BSD. However, the author did have a goal – to be able to take this from a fresh install to a "productivity system", complete with a GUI interface and accompanying tools. But along the way, we're aiming to see what's the same, as well as what's different.

And as we in the open source community know, variety is the spice of digital life. So we owe it to ourselves to explore this BSD system and see what's what.

CREDIT: The FreeBSD® logo is a registered trademark of The FreeBSD Foundation.

Why use FreeBSD? A better question might be "Why not use FreeBSD?" As we'll see, FreeBSD comes from the same Unix-y roots as Linux, and we can expect our skills to transfer over nicely. Oh, there will be slight differences for sure, but a shell is a shell is a shell, and your *ls* command in FreeBSD will indeed list the contents of your current directory.

But there's always something to be learned from different ways of doing things, and some of the areas where FreeBSD excels include:

› As a server. FreeBSD powers some of the most basic services on the internet, and is also used in some of the network hardware that makes up its infrastructure.

› Security. Its beginnings as a server OS have led to FreeBSD including a number of unique security features, such as access control lists (ACL), a choice of powerful firewalls, and jails for running virtualised OSes.

› Following documentation. With many modern Linux distros, getting installed is a simple point-and-click affair. In contrast, FreeBSD comes from an era where you actually had to read the documentation, understand it, and follow along closely.

› A focus on compiling source. The Ports system gives a fantastic introduction to compiling software for those who aren't software engineers by training.

Use FreeBSD for fun. Use it to learn, and evangelise it to others in the community – we're family after all. The way you install FreeBSD is just like Linux, in that you need to get some sort of installation medium, then boot your machine from said medium. The main differences during installation are threefold. First, the installer is text-based. Second, the installer doesn't give the option to move back to previous screens. Finally, the way you include software during the install is rather unique.

In this context, "include" software doesn't mean actually installing it. But the installer does give you the option to add the FreeBSD Ports Collection. Ports is the equivalent of Linux package repositories, except that they're used to build packages rather than install.

For now, if you've installed a Linux distro, you'll breeze through the text-based installer's screens:

1 First up, a couple of environment screens: Keymap selection, and choosing a hostname for your machine.

2 Select the "distributions". As mentioned, make sure you include Ports. Think of these as software sets, rather than the colloquialism for a Linux-based OS.

3 Disk partitioning. The first option will be between the traditional UFS and the newer ZFS filesystems. For testing, UFS is fine. You'll be offered the usual manually partition,or take over the whole disk. You'll also have a choice of several different partition schemes.

4 The installer will start decompressing the distro onto your drive. Don't go anywhere, it won't take long.

You may start to panic when you see FreeBSD's text-based installer. But its steps (like Partitioning as shown here) will be familiar.

5 Finally, some quick system configuration, including setting up the root user, network, time zone, date/time, services (SSH is selected by default), security options, and user account(s).

There are sub-steps, and it may seem like we glossed over the installation. But honestly, if you've done a Linux install you'll have no troubles. For those of us of a certain age, step 2 might even bring back fond memories of writing sets of Slackware floppy disks.

Booting into FreeBSD

That nostalgia will continue once you reboot your system. You'll get to see the plain-text system messages blur past in all their glory, and when all is said and done you'll land at a login prompt – no fancy desktop, no new-fangled session manager. You'll see `login:` and that's it. It doesn't even clear the screen for you.

But start poking around and things will begin to get familiar pretty quickly. Your home directory is in **/home/<username>/**, and the standard *sh* shell is similar enough to *Bash* that you won't notice the difference. The first thing that will probably jump out at you is that *sudo* isn't installed by default, so you'll need to use the actual root account to make changes. But otherwise all the basics are there: *cp, mv* for shuffling files, *mkdir* for new directories, and *cd* to get yourself around.

That said, your system is definitely barebones. Take a tour around your executable directories (**/bin**, **/sbin**, **/usr/bin**, and **/usr/sbin**) if you don't believe me. You've got basically enough to do some text editing (with either *edit* or <sigh> *vi*). In order to make this a useful system, we'll need to get started installing some programs.

Note: one area where the author ran into some delays is in setting up the bootloader. In part, this has to do with his stubborn insistence on trying to make it work with the native Windows boot manager on this dual-boot device. Things went much more smoothly once FreeBSD was allowed to take over the entire disk, and even just installing the FreeBSD bootloader would

» MEET THE BSD DEVILISH FAMILY

FreeBSD is similar to Linux in that there's an extended family of other systems. There's now quite a few different choices in the BSD world. Here are a few of the most popular:

› FreeBSD (**www.freebsd.org**) was originally a server OS first (as most Unix-likes were), but now is positioned as the "jack-of-all-trades" system, if you're willing to put in the work in, as we'll cover.

› NetBSD (**www.netbsd.org**) is another all-purpose OS, with a focus on running on as many different devices as possible, meaning you can just as easily find it on a desktop, a server, or replacing Windows CE on an NEC MobilePro 790 circa 2004.

› OpenBSD (**www.openbsd.org**) its claim to fame is security, with all packages going through rigorous audits at the source level.

› TrueOS (**www.trueos.org**), originally called PC-BSD, this is a desktop-friendly remix of FreeBSD, but pre-configured with niceties such as a graphical installer and bundles of apps pre-installed.

› DragonflyBSD (**www.dragonflybsd.org**) forked from FreeBSD and provides a carefully tuned system for high-performance applications.

› GhostBSD (**www.ghostbsd.org**) is technically a grandchild of FreeBSD, as it's built off of TrueOS. Just as Linux distros stand on the shoulders of giants, this aims to provide a simpler experience than its parent, with a focus on lighter desktops like MATE and Xfce.

likely have made things easier. It's worth noting that FreeBSD is planning on making direct UEFI booting available in a coming release.

Installing software

FreeBSD offers two ways to get software. The first is the Ports Collection. If you included it during your installation, poke around the **/usr/ports** directory. Alternately, you can opt to install software pre-compiled as well. Each of these has their own pros and cons.

Installing the pre-compiled packages is faster and potentially easier/less error-prone. However, building software on your machine will better tailor it for your hardware. The first step is to get the package tool set up on your system. Enter `pkg` in your shell, and if it's not already present you'll be prompted to download it. Your first step should be to update the package database:

```
pkg update
```

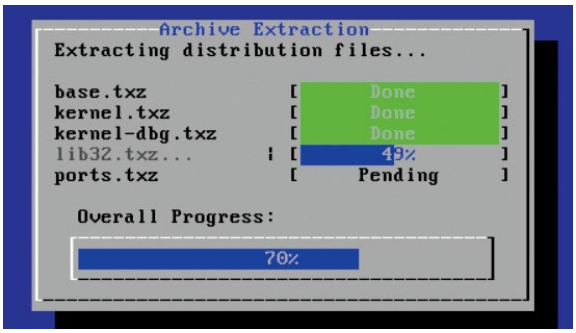
Then, search for your desired program with:

```
pkg search [your desired term]
```

To install something, use this:

```
pkg install [package name]
```

You'll see the progress of the install in the terminal, including the calculation of dependencies, the download of all that software and the actual installation. If you need to backtrack, it's just as simple:



The installer's "distributions" are package sets. You only have a choice of adding up to seven of these sets as part of the installation.

```
pkg delete [package name]
```

You can see all the packages you have installed with:

```
pkg info
```

Again, if you're a Linux user who regularly drops down a quick terminal to search and install software in Ubuntu or Fedora, it will take you all of 14 seconds to get the hang of this. And installing Ports is just as easy.

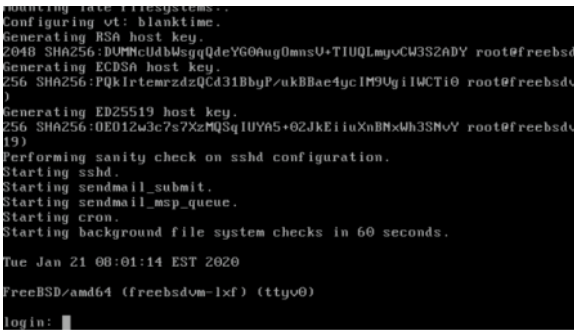
Day-to-day usage

Using FreeBSD on a day-to-day basis is precisely as described in the previous section. You'll find a lot of things very familiar as you're navigating around the console. And of course, applications operate just the same on FreeBSD as they do on Linux. There are only two areas where a little adjustment will be required. The first is system configuration. While the **/etc/** directory exists and does indeed contain many configuration files, there's a useful one that's unique to FreeBSD. As you work with FreeBSD you'll see references to making entries in the **rc.conf** file. This contains a smattering of important system-level settings. If you open it up, you'll see some of the things you set during install (such as network config). It's very short, but one would expect it to grow longer with your personal configuration.

The *bsdconfig* utility can help with some of these tasks, and it's as close to a control panel as you'll get without installing some sort of desktop that contains one. It's a menu-driven terminal program not unlike the installer, where you'll use arrow keys and Tab to jump around the dialogue. It won't be too much of an adjustment, as even some modern package systems will use these types of dialogues when configuring individual packages.

You should also expect to do a bit more troubleshooting on FreeBSD. To clarify, this isn't because things in FreeBSD are broken, but rather there are things that FreeBSD does differently by design that a Linux user will have to learn through research. A case in point: the *pkg* command does follow dependencies, but not quite in the way a Linux user might expect.

The author issued a quick `pkg install lxqt` command, to which the output indicated that a number of packages would be installed. On completion the `startlxqt` command failed with an error. Some quick checking revealed that the LXQt desktop was installed, but Xorg was not. When the startup failed again after the X install, further poking about showed that while the base X server was there along with the top-level LXQt desktop, the Qt5 toolkit that sits between them was still missing. And even when this was installed, using the `startx` command launched a default twm-based affair.



Before you log in, all the dmesg output will scroll by you furiously. Once it's complete you end up with a prompt at the bottom of the screen.

» FREEBSD'S LINUX COMPATIBILITY

While it may not be compatible with other BSDs, FreeBSD does come with a feature that enables you to run Linux software on it. The Linux Binary Compatibility layer allows you to install both 32 and 64-bit Linux applications, although it appears the support for i386-based software is fading. To add Linux compatibility, first add the Linux module for the FreeBSD kernel (while logged in as root):

```
kldload linux64
```

Then install the package (or build the port) for the emulation layer:

```
pkg install emulators/linux_base-c7
```

Add the following to **/etc/rc.conf/** to make sure Linux compatibility is loaded at start-up.

```
linux_enable="YES"
```

Your FreeBSD system is now set up to run Linux programs. Some other considerations you may need to take into account include:

› Does the program require any shared libraries? If so, you'll need to track them down and install them as well.

› Does the program come in a particular package format? Some traditional packages such as **.DEB** or **.RPM** are easy enough to crack open, but newer ones like *FlatPak* (see **LXF244** for a deep dive) or Canonical's *Snap* may not be as simple.

See the handbook (**www.freebsd.org/doc/handbook/linuxemu-lbc-install.html**) to learn more about these and other topics.

Manually adding an **.xinitrc** file with the following finally let `startx` bring up a usable LXQt desktop.

Again, this isn't to say what occurred was wrong. But based on past Linux experience, the author was expecting the installation to include all dependencies and the launch script to start everything up "automagically". But in FreeBSD, you need the wherewithal to install **X.org** yourself, launch it in the "standard" way, and make sure your environment it set to trigger LXQt when you do.

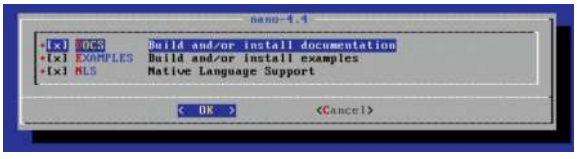
The designation for devices are similarly different. In FreeBSD, you reference the first internal drive with **/dev/ada0/**. Partitions are alternately referred to as **slices**, so the second partition may be **/dev/ada0s1/** or **/dev/ada0p1/** depending on where you're looking. Likewise, the default CD-ROM is represented by **/dev/cd0/**. Is using "ada" for the hard drive "wrong?" Who knows! Just understand that you may need to do some translation of nomenclature along the way.

FreeBSD is DIY

This speaks to the philosophy of BSD, which is to assemble the system you want with the software you want. In the course of writing this article the author frequented the FreeBSD message boards (**https://forums.freebsd.org**), on one occasion looking for hints on a GUI package manager. Firstly, the FreeBSD community is very active on forums, and give extremely accurate answers. But the answer to questions like "Where's the GUI package manager for FreeBSD" was a resounding, "We don't have one, we like *pkg* or *port*. Feel free to try some until you find one you like though." This wasn't intended to be dismissive, but rather to encourage the poster to build his/her platform to taste.

Yet the FreeBSD developers have taken pains to make the assembly easy where it makes sense. A quick confession: this author was fully prepared to hunt down **Xorg** configs (or **XF86Config** files, as those of a certain age will remember them) and tinker with modelines. At one time this was an extremely painful part of getting a Linux system running. But the **X.org** installation process on FreeBSD, the graphics system was configured, and a nice-looking desktop was the result. So, you may ask, why did the issue in the last section happen? Why didn't the whole desktop stack just install?

It's because FreeBSD didn't assume this author wanted to use **X.org** at all. Maybe Wayland was the intention. In this sense FreeBSD will often "do what you say, not what you mean". So you should be sure about what you're doing before you issue commands in FreeBSD, especially if you don't understand their function (e.g. there's a command that will happily re-build your entire system from source, likely occupying it for several hours) or target devices as mentioned in the previous section. In other words, understand the FreeBSD parlance for disk partitions before you start formatting drives.



Installing the nano text editor using the Ports system. It fetches the source for all dependencies, compiles, and installs them.

```
root@freebsdvm-lxf:~ # pkg install android-file-transfer
Updating FreeBSD repository catalogue...
FreeBSD repository is up to date.
All repositories are up to date.
The following 2 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        android-file-transfer: 3.9_1
        fusefs-libs: 2.9.9_1

Number of packages to be installed: 2

The process will require 1 MiB more space.
268 KiB to be downloaded.

Proceed with this action? [y/N]:
root@freebsdvm-lxf:~ #
```

Sometimes installing binary packages is the quickest and easiest way to get the software you want. And like Linux package managers, it is dependency aware.

Stated another way, think of modern Linux distributions as an office suite with easy-to-use buttons that sometimes don't get you precisely the result you want. FreeBSD, on the other hand, is *LaTeX*. You can get exactly the end product you want, you just also need to know exactly how to go about it.

Quickly, back to Linuxland!

This short journey into the world of FreeBSD tracked with what you'll find a lot in open source. It's easy to get started and achieve some measure of success and productivity. But as you start to dig into it, you quickly learn how far you still have to go.

Mentioning a couple of key take-aways at this point is prudent if you're thinking of giving FreeBSD a spin:

- Measure twice, cut once, aka do your research.
- The *FreeBSD Handbook* (**www.freebsd.org/doc/handbook**) is your best friend.
- Plan to do more things in the terminal than with Linux.
- FreeBSD often defaults to MVP (minimum viable product). You may need to manually add things that just "came with" Linux.

All that said, for Linux users a short holiday in FreeBSD is a great learning experience. And when it comes to the open source world, isn't that the point? LXF

» USING THE PORTS COLLECTION

First, navigate your way to the Ports Collection on your machine:

```
cd /usr/ports
```

A Port is basically a recipe for building the program in question. Spin through the various sub-folders and you'll start to see some old friends, such as desktops in the **/x11-wm** folder, the *Firefox* browser in **/www**, and (huzzah!) *Emacs* in **/editors**. In order to search for applications, we first need to index all the ports in the system:

```
cd /usr/ports
make index
```

Use the following command to search for something:

```
make quicksearch name=[name of your search target]
```

This will show you the name, location and a short description for your results. If you find what you want and would like to install it, you can use the command below. Just bear in mind you'll be compiling, not just installing, the application and all its dependencies.

To build and install a program, navigate to its Port directory and issue the below command. Note the **clean** parameter will make sure any temporary build files are deleted once it's done.

```
make install clean
```

Sit back and watch the messages scroll by, and after a few minutes/hours/days (depending on the size of the application), your hot-off-the-press program will be ready to run.

Uninstalling is just as easy (again, from the Port directory):

```
make deinstall
```

Roundup

Haiku » KolibriOS » OpenIndiana » ReactOS » Visopsys

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

Open operating systems

Using Arch with one hand while wearing a blindfold is no longer challenging for **Shashank Sharma**, who is now looking for something more arduous.

HOW WE TESTED...

Many of the OSes in this *Roundup* suggest that users first experiment with them inside the safe environment of a virtual machine. We heed their advice and run all of them inside VMs configured as per the specification mentioned on their respective websites, before subjecting them to physical hardware.

Our intention is to hunt for an alternate OS that enables us to use the computer productively. Sure, we don't expect to be able to stream games via *Steam* or watch movies via Netflix, but if they can play multimedia, browse the web, do word processing and such without throwing unexpected errors, they'll be in our good books. We'll test various parameters such as installation and app support, with the sole intention of helping us zero in on an alternate OS that takes us out of our zone of familiarity – but with the least amount of compromise with our comfort.

Being a regular reader, you'll know full well that Linux isn't the only open source operating system that you can pluck from the internet for free. Notwithstanding our love for Linux, we do regularly review other open source OSes as well, particularly the BSDs.

Many non-Linux open source OSes are designed for niche uses and wouldn't be of much use outside their specialised environments. While they might not possess the dexterity of Linux, which can scale from your netbook to a server farm, a few of them perform quite well in multiple environments, including the desktop. We'll introduce you to some of the best options for everyday desktop tasks in this Roundup.

Despite the fact that they've been under development for over two decades, many of these alternative OSes are actually in the early stages of development! This is primarily because they don't get the same attention from the open source ecosystem as most Linux distributions. Many of the OSes in this *Roundup* have a very limited number of core developers (and an even smaller band of dedicated users), and none of them have the backing of multinational for-profit corporations to accelerate their development – yet they continue to grow.

Installation

What does it take to anchor the OS to your machine?

Almost all the OSes in the *Roundup* are equipped with a graphical installer. Unfortunately, though, that doesn't mean you can expose them to a physical drive on a production machine as soon as you download the OS. You should realise that not all of them have undergone the same amount of testing as the Linux installers – and as we mentioned, it's always a good idea to test in a VM first.

The one exception is OpenIndiana. Linux users won't have any trouble navigating through its installer, which is very polished and streamlined, much like a typical Linux installer. You can use the partition built-into the installer to carve space for the OS, or use the familiar *Gparted* tool that's available in the Live environment.

Then there's ReactOS, which is the only OS in the *Roundup* which offers a dedicated installable edition. Also available is a Live ISO image, designed to help you test the OS on your hardware. Once you're satisfied that everything works, you can use the install-only variant to anchor ReactOS to the computer. The OS's installer is a throwback to old Windows installers and isn't particularly difficult to use. Like many Linux distros, ReactOS runs through a first-boot configurator to help you create a user and select a theme for the desktop.

Haiku and Visopsys both have simple and straightforward installers. All you need to do is point Haiku to a BeFS filesystem partition and it'll copy the files and install the bootloader fairly quickly. The installer can now boot from GPT partitions and the 64-bit image ships with an EFI bootloader so that Haiku can boot on EFI computers.

Many OSes such as Haiku include a partitioner to help you create a dedicated partition for the OS.

The Visopsys installer also includes a partitioner to help you create a slice for the OS. You are forced to use the system as an admin user, but the installer enables you to set up a password for it. If you're installing Visopsys on a new virtual disk, make sure you choose the option to write the basic MBR to the disk in order to boot into the installation.

The one OS that frees you from the cycle of installation is KolibriOS, which doesn't have an installation procedure as such. All you need to do is to point *GRUB* to the **kolibri.img** file by adding an entry in the the **40_custom** file under **/etc/grub.d**, and you're good to go.

VERDICT

HAIKU	**8/10**	**REACTOS**	**8/10**
KOLIBRIOS	**8/10**	**VISOPSYS**	**6/10**
OPENINDIANA	**9/10**		

Besides OpenIndiana, we'd suggest you first install these OSes inside a VM.

Default apps

Are they usable out of the box?

Just like the OS it emulates, ReactOS includes a handful of apps. Besides a number of utilities including *Paint*, *Notepad* and *WordPad*, the OS includes the three favourite Windows card games.

Visopsys isn't much better and ships with a minimalistic suite of apps, since the goal of the OS is to create a fully functional OS for CS students and alternate OS enthusiasts like us. Besides the disk manager and the installer, there's a basic text editor, an image editor, a virtual keyboard app, a couple of games and a handful of administration utilities.

By contrast, KolibriOS ships with dozens of apps. There's an audio player, a video player, a VNC viewer, a text editor, a rudimentary web browser and more. The OS also has lots of tools for developers, particularly for FASM assembly, and several game console emulators.

Haiku too includes quite a lot of nifty but essential apps which a typical user might need on the desktop. There's an email client, a web browser, a media player, an image viewer and a text editor, as well as administrative tools like an activity monitor, a hex editor, a disk partitioner, a simple web server and more.

OpenIndiana ships with the all-too-familiar MATE desktop and a majority of the apps are from MATE's stable as well. Besides

KolibriOS is an absolute treasure trove for retro gaming aficionados.

these, there are a handful of mainstream productivity apps such as *Firefox*, *Thunderbird* and *Pidgin*. Straight out of the box, the OpenIndiana installation has limited usability and you'll need to use its package management system to flesh it out.

VERDICT

HAIKU	**8/10**	**REACTOS**	**4/10**
KOLIBRIOS	**6/10**	**VISOPSYS**	**4/10**
OPENINDIANA	**6/10**		

Besides Haiku, all other OSes leave out essential apps for everyday use.

Usability and performance

Does it take much effort to get used to them?

This *Roundup* includes operating systems that are called 'alternative' for a reason. They are unlike your everyday desktop distro and in fact it'll be unreasonable to expect them to look, feel and behave like other mainstream OSes as well.

Many of them are also in the nascent stages of development, despite the fact that developers have been working on some of them for over a decade. The early development stage is a reflection of the OSes' unstable base, which will have a direct impact on their usability and performance.

Even when it comes to mainstream distros, usability is an important factor for consideration. Having a long list of features comes to naught if a distro isn't usable. The same is true for these alternate OSes.

This is why in addition to comparing them on various individual aspects that influence their usability, we'll also rate their overall user experience to help you find one that gives you the most mileage on the desktop.

Haiku 8/10

Don't let Haiku's beta status fool you: the project is pretty mature and been in existence for over a decade now. The usability of the OS is phenomenal, both inside a VM and on real hardware.

The ISO gives you the option to either launch the installer or boot into a fully functional Live environment. The installer is very efficient at its job and also includes a partitioner to help you carve space for the OS. Even on a moderately powered machine, Haiku's boot-up and shutdown times are pretty impressive.

The OS works straight out of the box on most machines with standard hardware. It has one of the most comprehensive set of default apps, and you can fetch more using the graphical package manager. The user experience is further enhanced by the rich documentation and the project's support infrastructure, which is designed to habituate first-time users.

KolibriOS 6/10

KolibriOS is written in the FASM assembly language and based on the source code of the MenuetOS operating system. It boots in a flash and gives you access to a number of useful apps. There's no installation involved, though you'll have to make sure you select the option to save the changes you made during the session when you shut it down.

The OS supports FAT and NTFS filesystems and ships with drivers for popular audio, video and Ethernet hardware. The desktop is fairly intuitive to operate, and new users aren't reprimanded for casually exploring the desktop and its various apps.

The only shortcoming of the OS is that its productivity apps aren't really mature enough for everyday use, and the lack of a package manager doesn't help its case either. On the other hand, if you like retro gaming there's no better OS.

Documentation and support

Will they guide you through the murky waters?

Because Visopsys is primarily designed for developers and students, the bulk of its documentation caters for this group. There's a lot of information that exposes the internals of the OS, which is a treasure trove for any Computer Science student. KolibriOS also has just enough documentation to help users get started with the OS. There's a quick FAQ and some how-tos to help boot KolibriOS alongside Windows and Linux, as well as on virtual hardware. There's also a fairly active multilingual forum for dispensing help.

OpenIndiana fares a little better. It has a detailed FAQ and an (under construction) handbook, which is still fairly detailed. There's also a wiki to help users and developers get orientated with the OS. The project surprisingly doesn't have a forum, so you'll need to use its mailing lists for your support queries, which is a rather clunky way of doing things.

On the other hand, the ReactOS project has a wiki which hosts various tutorials designed to help first-time users install the OS and perform various tasks. The wiki is also home to developer documentation along with details about the internals of the OS. For support there's a MatterMost-powered chatroom and multilingual forums.

Similarly, Haiku's website does everything it can to help orientate first-time users of the OS. It has various introductory documentation including a detailed FAQ, a fairly complete illustrated user guide, along with several other focused guides to help users install, update and virtualise the OS. There are several avenues for support as well including forums, IRC channels, mailing lists and so on.

VERDICT

HAIKU	8/10	REACTOS	8/10
KOLIBRIOS	6/10	VISOPSYS	4/10
OPENINDIANA	6/10		

New users will feel most welcome with the support infrastructure of the ReactOS and Haiku projects.

OpenIndiana 6/10

Only available for 64-bit computers, OpenIndiana has separate images for optical media and USB devices. The images boot into a Live environment, from where you can launch the graphical- or text-based installer, which works just as expected.

You don't get the same 'alternative OS' feeling with OpenIndiana as you do with the other OSes on test here because of the use of the MATE desktop environment, as well as a healthy dose of marquee open source apps. The one major drawback as far as the usability of the OS is concerned is the lack of a graphical package manager. While the CLI version isn't hard to master, it'll alienate a good number of first-time users who are already out of their comfort zone in an alien OS.

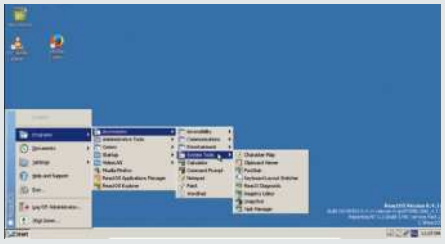
The lack of built-in documentation doesn't help either, but OpenIndiana is the closest you'll get to a typical Linux install.

ReactOS 8/10

ReactOS is an open source operating system based on the design principles of Windows NT. It's written completely from scratch and exists to give users an open source platform to run software designed for Windows, by being binary-compatible with the proprietary OS.

The project offers a Live CD to help explore the OS and an install CD with an easy-to-navigate installer. Whether you've used Windows or not, you'll find ReactOS intuitive to operate. Sure, it doesn't ship with many apps by default, but you can flesh it out with the package manager.

Besides physical hardware, you can also install ReactOS on virtual hardware, which is in fact a speciality of the OS given its alpha status. Despite its rather miniscule requirements, ReactOS only supports a limited set of devices and peripherals. When it does work, the biggest hindrance is the occasional Blue Screen of Death.



Visopsys 6/10

Considering the fact that it's primarily an educational system, Visopsys surprised us with its usability. What we thought of as an ideal candidate for the 'Also Consider' box surprised us with its features and performance. The OS boots in a snap and gives you the option to either launch the installer or boot into a Live session. The inclusion of a very capable partitioner in the installer is another pleasant surprise.

The desktop is fairly simple to navigate and includes a handful of essential apps, plus handy administration and configuration utilities. But attempts to use it for regular desktop tasks won't take you very far, as the OS is missing a web browser and there's no package management to help you pull in additional apps and utilities. Visopsys does a nice job of masquerading as a regular desktop, but it really is a very capable CS project and should be treated as such.

Hardware support and requirements

Will they even work on your computer?

Another aspect that keeps these OSes from becoming mainstream is their lack of hardware support. On the other hand, on compatible hardware most of them will outperform your favourite Linux distro.

For instance, Visopsys can power a graphical desktop from any Pentium computer with 64MB of RAM. The OS supports all variations of the FAT filesystem and has read-only support for ext4. In terms of peripherals it supports PS/2 and USB input devices, IDE and SATA disks, and PCnetEthernet cards.

Similarly, you can run ReactOS on any Pentium-class x86 machine with just 256MB of RAM. ReactOS collaborates with various other projects such as Haiku for its hardware support, which is still fairly limited in comparison to some of the other OSes in this roundup.

KolibriOS isn't much different either. It's a 32-bit only OS and requires miniscule amounts of resources. Its developer claims it'll even run on machines with the original Pentium processor. The OS supports USB 2.0, and quite a few popular Ethernet devices as well as Radeon and Intel graphics drivers. OpenIndiana needs a 64-bit computer with at least 4GB of RAM for decent performance. The project also hosts a community-maintained list of supported peripherals, which is fairly detailed.

Haiku produces images for both 32- and 64-bit machines. The project borrows its Ethernet and Wi-Fi drivers from FreeBSD, via a KPI compatibility layer. They've been upgraded to those from FreeBSD 11.1 and support the Atheros 9300-9500 families, Intel's newer 'Dual Band' family, some of Realtek's PCI chipsets, and newer-model chipsets in all other existing drivers.

VERDICT

HAIKU	8/10	REACTOS	6/10
KOLIBRIOS	8/10	VISOPSYS	5/10
OPENINDIANA	8/10		

While they'll run on your everyday computers, don't expect these OSes to support all your hardware.

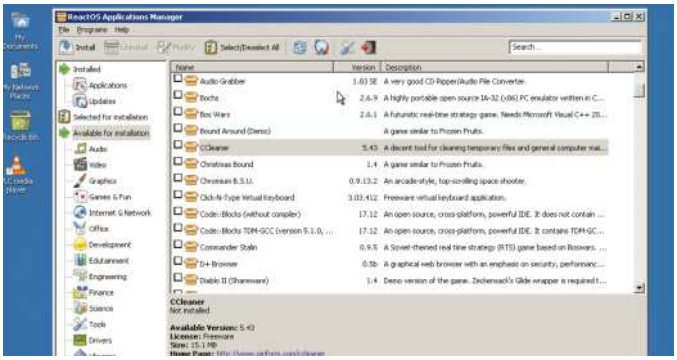
Package management

What apps are available in their repositories?

Some of the OSes in this *Roundup* ship with a lot of apps, but others are rather bare-bones and aren't really useful straight after installation. You'll have to flesh these out to use the OS productively on the desktop. Unfortunately, not all of the OSes let you install additional apps, and some take far more effort than others.

KolibriOS and Visopsys belong to the former category: there is no means of installing additional packages to them. While OpenIndiana does have a package management system, and quite an adept one at that, fleshing out the OS is one of the weakest points of the OS. The OS uses the Image Packaging System but features a CLI package manager, *pkg*, rather than a graphical one. While *pkg* is very similar to *apt* and *dnf*, you will have to read through the *man* page and other documentation for tasks such as adding repositories.

On the contrary, package management is the only saving grace of ReactOS's vanilla installation. The OS uses a *Synaptic*-like app store that appears when you use the Add/Remove Programs function in the Control Panel. The ReactOS Application Manager is intuitive to operate and enables you to install multiple programs in bulk. You can also continue to use the app while it's downloading and installing software. Similarly, one of the highlights of the latest release of Haiku is its graphical package manager, *HaikuDepot*. It's intuitive to operate and works pretty much like the package management systems on Linux.



Besides apps like LibreOffice and Firefox, you can also use the ReactOS Application Manager to download libraries like Adobe AIR and Microsoft .NET framework.

VERDICT

HAIKU	8/10	REACTOS	8/10
KOLIBRIOS	0/10	VISOPSYS	0/10
OPENINDIANA	5/10		

Being stuck with the default apps isn't as much of an issue in KolibriOS as it is in Visopsys.

Daily use

Can you use them as your everyday desktop?

You can definitely use OpenIndiana as a regular desktop, especially when you consider the fact that it ships with a familiar environment and mainstream open source apps. However, the OS doesn't include all the apps you'll need and you'll have to fetch additional ones using its CLI package manager, which interrupts an otherwise smooth user experience.

Although Visopsys is a fully functional OS, you can't really use it as an everyday desktop in the traditional sense. The OS is developed by an individual and is more of an educational tool rather than an actual OS which you can put to use for everyday tasks. It also lacks crucial productivity apps such as an internet browser and a package management system.

ReactOS is discounted because, by its own admission, the project isn't yet ready for use as your daily desktop. It won't disappoint you in terms of app availability, but the system isn't stable enough yet to be used reliably for productive work. The Live CD only worked on one of our test machines, and installing the *VirtualBox* Additions brought up the infamous Windows-like Blue Screen of Death.

In contrast to the previous two, KolibriOS ships with a large number of apps out of the box. Unfortunately, many of the crucial ones are rather rudimentary. For instance, the web browser is text-based, and while it had no issues rendering simple sites like the KolibriOS website, it crashed whenever we tried to use it to visit data-intensive websites such as YouTube. The one task we absolutely recommend the OS for is retro gaming, for which it remains unparalleled.

The Time Slider utility in OpenIndiana takes advantage of the ZFS filesystem to take snapshots of the installation at timed intervals.

All things considered, the OS you'll be most comfortable using for daily desktop duties is Haiku. It has impressive hardware support and doesn't throw any unexpected errors during installation. The collection of apps on the desktop, such as the WebKit-powered web browser and the availability of useful apps in the package management utility, surely helps it meet the desktop computing requirements of a large number of people.

VERDICT

HAIKU	8/10	REACTOS	5/10
KOLIBRIOS	5/10	VISOPSYS	4/10
OPENINDIANA	8/10		

OpenIndiana and Haiku are your best bets if you're looking to be productive.

Open operating systems
The Verdict

This is an interesting *Roundup* to rate. Visopsys brings up the rear, but it doesn't lose out in the traditional sense of the word. The OS is in fact in a league of its own and has no peers. Instead of looking at it as an alternative OS which you can use productively, think of Visopsys as a computer science project on steroids. If you need to wrap your head around the inner workings of a multi-threaded, fully preemptive multitasking OS, there is no better option than Visopsys.

While ReactOS's default desktop doesn't offer much, you can quickly turn the vanilla installation into a productive one thanks to its software repository. Unfortunately, fleshing out the OS with useful apps doesn't automatically make it usable. The alpha nature of the OS and its limited hardware support means that it will continue to operate in the confines of a virtual environment for the time being.

KolibriOS makes it to the podium, but in last place. Straight out of the box, it offers all of the apps you'll need on a desktop for everyday use. Unfortunately, however, several crucial ones – such as the web browser – are quite rudimentary, and you'll be testing their limits in no time. But as we've said before, KolibriOS makes a wonderful option for anyone into retro gaming.

If an AI (that's *exactly what an AI would say!–Ed*) was doing this *Roundup*, it would have awarded the win to OpenIndiana, as it scores highly on all the parameters except one. It's the only OS in this Roundup that looks and feels like any other Linux distro featuring the MATE desktop. Complementing the familiar-looking desktop are commonly used productivity apps for everyday use. The weakest point of the OS is the lack of a graphical package manager, which might make new users hesitant about venturing into this OS.

Ironically, it's precisely that lack of an alien feeling that keeps OpenIndiana from claiming the top spot on the podium: it looks too familiar to be considered as an alternative OS. All things considered, Haiku has the right mix; it looks like an alternative OS, behaves like one too and yet enables us to be productive enough to use it as a daily desktop.

1st Haiku 8/10

Web: **www.haiku-os.org** **Licence:** **MIT and others**
Version: **R1/beta1**
The alternative OS which gives you the best mileage.

2nd OpenIndiana 8/10

Web: **www.openindiana.org** **Licence:** **CDDL and others**
Version: **0.4.11**
An alternative for those looking for familiarity.

3rd KolibriOS 6/10

Web: **http://www.kolibrios.org** **Licence:** **GPL v2**
Version: **2018.10**
Use it for its retro-gaming emulators, if nothing else.

4th ReactOS 6/10

Web: **https://www.reactos.org** **Licence:** **GPL, LGPL, BSD**
Version: **r7617**
A Windows look-alike that is most usable on virtual hardware.

5th Visopsys 6/10

Web: **http://visopsys.org** **Licence:** **GPL, LGPL**
Version: **0.83**
A wonderful option for CS students and OS enthusiasts.

» ALSO_CONSIDER

There are several other open source OSes that we didn't include for one reason or the other. The most mainstream alternative is TrueOS, formerly known as PC-BSD, which is a desktop flavour of FreeBSD. There's also GhostBSD for those looking for a MATE-based desktop.

For fans of AmigaOS there's Syllable Desktop, which wasn't included as it hasn't been updated for several years. Similarly, it's been some years since the last release of MenuetOS, which is the progenitor of KolibriOS. The 64-bit version of Menuet had a release last year, but that version isn't open source.

Licence ambiguities also prevented us from featuring another AmigaOS clone, Icaros, which is based around the AROS Research operating system. We also couldn't spot the source code repository. There are other AROS-based OSes, such as AspireOS and AROS Broadway, but they either have severe hardware restrictions or aren't actively maintained.

Roundup

Equinox Desktop Environment (EDE) » Lumina » LXQt » Moksha Desktop » Openbox

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

Lightweight desktops

With his gym membership going unused, **Shashank Sharma** decides to cut down the flab on his desktop instead.

HOW WE TESTED...

We installed all the desktops inside an Arch installation. Most of them are in one of the officially supported repositories, though some are in the community-powered Arch User Repository (AUR).

We've rated the desktops on various fronts, such as their availability and cache of default apps. Malleability is also another important criteria when choosing a desktop; an environment that ships with loads of tweakable settings will score higher than one that has a limited number of modifiable parameters, as people usually swap their defaults DEs because of a rigid configuration. Just as important is the availability of plug-ins and add-ons. These give you the option to extend the usability of your environment depending on your requirements.

The most important criteria for this *Roundup*, however, is the usability of the desktop. Any desktop that doesn't fare well in this test wouldn't be of much use, despite achieving high scores on other fronts.

Initially at least, it'd seem counter-intuitive that anyone would want to change the default desktop environment (DE) of their distribution. After all, the leading distributions spend a considerable amount of time fine-tuning the user experience. Some are even actively involved in the development of the DE to ensure that it meets their expectations. However, the default environment, pretty much like the factory installation, is designed to suit most common workflows.

Despite all their refinements and functionality, some DEs are just a hindrance, especially for tasks that need the least amount of graphical oomph. Then there are those of us who have interacted with computers all their life with a DE that conforms to the traditional desktop metaphor, and so don't really see the advantages of newfangled desktop shells. Another group of users disgruntled with the default DE are those computing on under-powered machines which lack the resources to satiate the needs of the resource-guzzlers.

If you identify with any of the use cases we've just outlined, this *Roundup* is for you. We'll look at some desktops that have a minimum footprint but will still enable you to get along with your tasks.

Availability

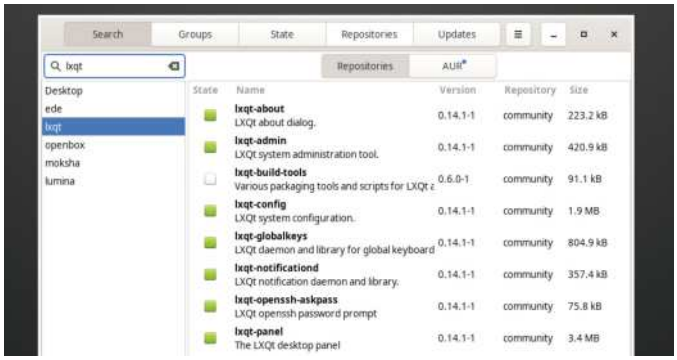
Can you find it in a repository near you?

With the exception of a couple, all the desktops in this *Roundup* are available in the repositories of mainstream desktop distributions. Arch, however, leads the pack, in that it gives you access to all the options via its official and community-supported repositories. You'll also get good mileage with Fedora, thanks to its Copr build system which complements its repositories.

Similarly, Ubuntu with its PPAs enables you to get your hands on a wide number of DEs. One word of advice though: the quality of packages in the community-maintained repositories varies widely and ranges from anywhere between rock-solid to barely usable. You might not get the best experience using packages from a community-supported repository as compared to the official ones.

LXQt and Openbox are the two options that are universally available in almost every Linux distribution. LXQt maintains an extensive list of projects it supports officially on its website, and you can install it on Fedora using a Copr repository. Openbox too is universally supported. We couldn't bring up the right-click applications menu using the package in the repositories of Fedora 29, but that was easily (*oh really?–Ed*) resolved by downgrading the python-pyxdg package.

Lumina is primarily developed for the BSDs, but has also been ported to some Linux distributions. The project's page lists a few distributions that have pre-built packages, including Debian, Gentoo and Arch. The desktop is also available via a Copr repository for Fedora, but it doesn't perform well and crashed during virtually every session. Ubuntu users can find guides to take them through the process of compiling it from source, which isn't too cumbersome.



We'll advise against installing several desktops inside a single installation. Not only will it needlessly clutter the application menus, some don't play well with the others.

Equinox Desktop Environment (EDE) hosts installation instructions on its official wiki and covers Arch, Debian, Slackware, Alpine and a couple of BSDs. The project itself lists these packages as experimental and advices you to compile EDE from source for the best user experience.

The Moksha desktop is a fork of the Enlightenment (E17) desktop. Again, it's the default only on Bodhi Linux and we couldn't get it to work on any other distribution besides Arch. There is a Copr repository but its last build failed for Fedora 25.

VERDICT

EQUINOX DESKTOP	**5/10**	**MOKSHA DESKTOP**	**5/10**
LUMINA	**7/10**	**OPENBOX**	**8/10**
LXQT	**9/10**		

Desktops designed for a particular distribution don't work well on others.

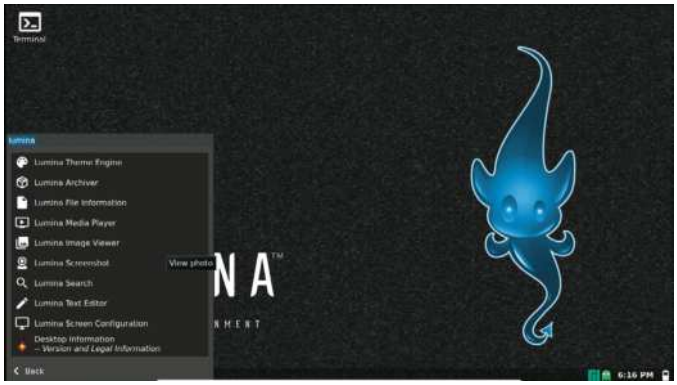
Default apps

What's on offer?

One of the ways some desktops help you conserve resources is to offer matching lightweight apps for common desktop tasks such as file management, text editing, image viewing and so on. Others leave it to the user to pair the lightweight desktop or window manager with third-party lightweight apps.

EDE belong to the latter category. The goal of the desktop is to provide a desktop environment only, and you will have to add even the most common apps like a file manager. Openbox similarly concentrates on being a lightweight window manager that you can complement with third-party light apps. It's the same story with Moksha. Since it was designed to ship with Bodhi Linux, the desktop borrows the few essential apps that are bundled with the distribution.

The other two options are a lot better in terms of bundled apps. Admittedly, you won't find apps such as web browsers, email clients and office suites in Lumina, but the desktop does bundle a handful of utilities that are written specifically for the project. These include an archiver, a scientific calculator, a configurator, a PDF viewer, a text editor, a media player and a file manager. LXQt too features a small cache of optional apps which add extra capabilities to the desktop. There's an image viewer and screenshot tool, a lightweight terminal emulator, a file archiver, a process manager, a file manager and desktop icon manager – which is essentially just a Qt port of *PCManFM* – and more along those lines.



Lumina has a good collection of lightweight apps which nicely complement the desktop itself.

VERDICT

EQUINOX DESKTOP	**1/10**	**MOKSHA DESKTOP**	**2/10**
LUMINA	**7/10**	**OPENBOX**	**1/10**
LXQT	**8/10**		

You'll like Openbox's clean slate if you want to build your own environment.

User Experience

Are they a boon or a bane?

Lightweight or not, a desktop environment is one of the most crucial elements of a distribution. Regardless of the features or application set with which a distribution ships, oftentimes it's the look and feel, behaviour and responsiveness of the desktop environment which help you decide on a distribution.

Perhaps this explains why most mainstream distributions spend a considerable amount of time polishing the desktop environment and ensuring a rich and rewarding user experience.

But all of those careful design strategies go out the window when you swap the default environment with a lightweight alternative. That said, while the primary objective of these desktops is to save resources, it doesn't mean they totally ignore the usability aspect.

After all, no one wants to use a desktop that takes a negligible amount of memory but which is extremely cumbersome to operate.

Equinox Desktop 7/10

EDE isn't the easiest desktop to install. If you aren't running Arch, chances are you'll have to manually compile it from source.

The desktop has a retro feel to it and starts with an old-school splash screen. EDE adheres to the classic desktop metaphor, with a status bar at the bottom of the screen and the applications menu at the extreme left. You use the text box besides the applications menu button to run a command, which can also be used to quickly launch apps. There's also a pager to switch desktops.

The right-click context menu offers the traditional options and helps place icons on the desktop and change the wallpaper. You can tweak other settings with the desktop configuration app, which provides a bunch of tools to customise other basic elements. Advanced options can only be set by editing text files, which robs the desktop of some usability points.

Lumina 7/10

From our experience, getting this desktop to work on Linux is a really cumbersome process. It's unstable and behaves abnormally atop most distributions, with frequent crashes and weird graphical artifacts pasted across the desktop.

Instead of a traditional splash screen, Lumina instead displays a quote. The desktop has a clean layout, with a status bar at the bottom. The right-click context menu has a number of useful options, including an applications menu. This is a good thing™ because the default applications menu takes a little getting used to. In the default view it only displays favourite apps, and it takes an additional click to get to the full list of installed apps.

There's also a textbox to help you zero in on the app you wish to launch. Besides the apps, you can also launch the file manager and the Lumina configuration utility from this menu.

Help and support

What to do you when you're stuck?

EDE's official wiki has instructions on how to compile it for various desktops – but the majority of the documentation is geared towards helping developers contribute to the project. Also, unlike some of its peers in this *Roundup*, EDE gets very little coverage on other websites besides its own, primarily due to its slow rate of development.

Lumina is the default desktop of TrueOS BSD, but has been ported to various other BSDs and Linux distributions. Despite having a website of its own, the desktop offers little information to help you get started. The project engages with the community using the Lumina Desktop Telegram channel. Besides these, there are no avenues for seeking help, so If you're stuck it's best to take the issues to your distribution's forums.

Moksha is the official desktop of Bodhi Linux but has been ported to other distributions as well. The desktop doesn't have a support infrastructure of its own and shares the one from the larger Bodhi Linux project. By contrast, LXQt boasts ample documentation and avenues for engaging with the user community should you need assistance. There are mailing lists for both users and developers, as well as fairly active forum boards. There's a wiki which covers installation of the binary packages for various distributions as well as instructions for compiling the desktop from source.

Similarly, given Openbox's age the project has extensive documentation and support from both official and unofficial sources. The project's website is a wiki with loads of information to help orientate new users. If you're stuck, there's a high probability that your distribution has an official support channel.

VERDICT

EQUINOX DESKTOP	4/10	MOKSHA DESKTOP	4/10
LUMINA	4/10	OPENBOX	8/10
LXQT	8/10		

If you're using a lightweight desktop on your regular distribution, go with the one with active support options.

LXQt 8/10

This desktop environment is a combination of the GTK-based lightweight desktop LXDE and Razor-Qt, which was an equally lightweight, but far less mature, desktop that used the Qt toolkit.

Thanks to this combination, LXQt manages to pull off the look and feel of a modern desktop without being a drain on resources. You can find LXQt in the repositories of virtually all distributions, and if your favourite distribution has a version for under-powered machines, chances are it's running LXQt.

It adheres to the old but familiar desktop metaphor, with a status bar laden with icons at the bottom of the screen. The applications menu features the traditional categorised list of apps as well as a search box to help launch apps. LXQt offers a decent number of tweakable options that help customise the most commonly used aspects of the desktop.

Moksha Desktop 7/10

This is a fork of the Enlightenment 17 desktop and features Bodhi Linux-specific changes, which the developers have been patching into E17's code over the years.

Moksha is fairly intuitive and easy to customise. It employs tons of visually appealing gadgets to display system information. These gadgets can be placed on the desktop as well as the taskbar, or 'Shelf' in Moksha speak. The desktop can be completely controlled via keyboard and you can define your own keybindings. You can also use the settings panel to influence the look and feel of your installation.

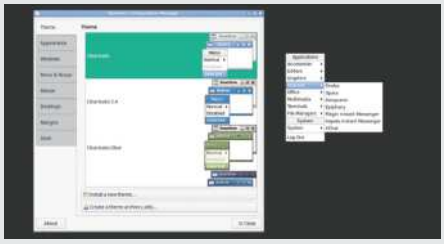
However, unlike E17, Moksha isn't readily available in the repositories of many distributions. Another issue with the desktop is its applications menu. While it does give you access to other areas of the installation besides applications, it doesn't feature a search box which can be used to quickly launch applications.

Openbox 7/10

Openbox is readily available in the software repositories of most distributions. Its minimalist yet customisable nature make it a robust choice for experienced users.

Its window manager is in fact so barebones that you wouldn't even notice it's there. All you get, by default, is a wallpaper-less background and a cursor. An application menu only appears in the right-click context menu. You can use the menu to launch apps, which run within windows with the usual controls and behave as you'd expect in any desktop.

The window manager is often combined with various other lightweight elements on the desktop. For instance you can use either *Tint2*, *Plank* or *Docky* to create a taskbar, and *PCManFM* as the file and desktop manager. You can also customise different aspects of the window manager using the *obconf* tool. Extensive manual configuration may not appeal to all users.



Performance

Do they actually perform better than the fully fledged defaults?

It's very difficult to measure performance in terms of absolute numbers due to a lot of factors. Since we can't separate the desktop environment from the apps, we can't really measure their exact draw on resources. This means the consumption will vary from machine to machine and may go up or down depending on the number of installed apps.

Also, there is no one single point of measurement. For instance, an environment could be blazingly fast to load, but its resource consumption could be really high while it's supposedly idle because of the fact that it's prefetching or loading components in the background.

On our test machine, EDE idles at about 150MB. The figures go up when an app is launched, but quickly drop back when you close the app. On a related note, it's fairly quick at launching apps, with *LibreOffice* timed at 4.3 seconds.

Then there's Lumina, which isn't designed for Linux – and that's probably why its startup takes about three times longer than EDE. It idles at around 210MB, but app launch times are only slightly slower than EDE. Its memory consumption also mirrors EDE, with highs recorded at the initial launch of apps.

Moksha is just as lightweight as EDE but its startup times and app launches are noticeably faster. While LXQt does take slightly longer to bring up the desktop, its app launch times are on par with Moksha.

Openbox is blazingly faster than the others. The startup time for the window manager depends on the number of elements it has to load. Once it's loaded, app launches are among the fastest of the lot.

VERDICT

EQUINOX DESKTOP	8/10	MOKSHA DESKTOP	9/10
LUMINA	8/10	OPENBOX	9/10
LXQT	8/10		

There's a reason why the desktops featured in this Roundup are considered lightweight, and it's not because of the disk space they require.

Plug-ins and extensions

Going above and beyond.

Just because these desktops are lightweight doesn't mean they can't provide the bells and whistles you get with mainstream desktops. As these elements inevitably increase the memory footprint of the desktop, some of the desktops bundle them inside a separate package, while others avoid them altogether.

EDE doesn't believe in offering any additional plug-ins and so you're stuck with the default functionality. On the other hand, while Openbox can be extended by plugging in various components such as a file manager, the window manager itself cannot take on new features via plug-ins, extensions or additional modules of any sort.

Lumina began life as a set of extensions to Fluxbox, a stacking window manager for the X Window System, and bills itself as having a plug-in-based interface design. This enables you to customise the desktop as needed simply by choosing which plug-ins to have running on the desktop and taskbar. But the process to change the extensions isn't very apparent, and doesn't always work if the desktop is installed on distributions besides TrueOS.

Then there's LXQt, which uses the concept of modules – essentially desktop-independent tools that operate as daemons for the local user on desktop specific operations. The desktop is further made up of several optional components such as the display manager and the power management module. LXQt's panel also supports plug-ins.

Moksha offers plenty of add-ons which you can plaster all over your desktop.

Moksha is a treasure trove of extensions, plugins and modules that add functionality to the window manager itself. It also has gadgets that are placed on the Shelf or the Desktop.

VERDICT

EQUINOX DESKTOP	**N/A**	**MOKSHA DESKTOP**	**8/10**
LUMINA	**3/10**	**OPENBOX**	**1/10**
LXQT	**6/10**		

Besides Equinox and Openbox, all these desktops enable you to further extend their functionality.

Configurability

Mould them as you like.

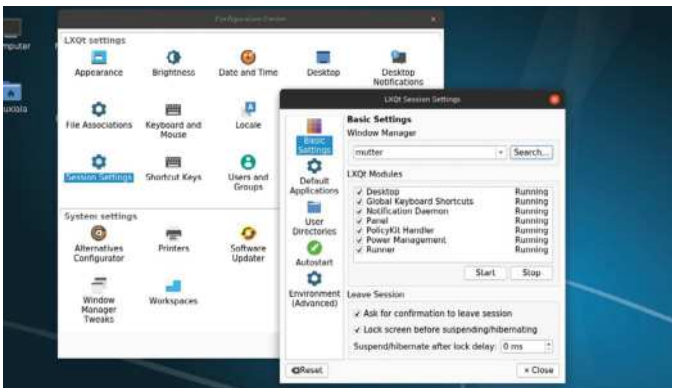
Since you've taken the steps to add a new desktop, it's only natural you'll want it to do your bidding by customising the way it works.

All configuration options in EDE are housed inside the configuration window. It provides six tools to customise different but very basic aspects of the desktop including wallpaper, screensaver, calendar, volume and preferred apps. Advanced users can also set various other aspects of the desktop by manually editing configuration files.

Openbox users can customise the window manager using the Openbox Configuration Manager (*obconf*) app. This helps you customise little else besides various aspects of the window manager. For instance, you can use it to change the default appearance and apply themes, and change the behaviour of the mouse and the position of the taskbar.

You can use Lumina's configuration utility to tweak various aspects of the desktop, such as the theme and window effects. You can also use it to set up default apps and define keyboard shortcuts, as well as alter the layout of the three critical aspects of the desktop – namely the desktop, panels and menu. While most of the tweaks apply system-wide, there are a couple that are restricted to the logged-in user.

LXQt's configuration centre offers similar options, albeit in a much nicer interface. But this is to be expected, as, along with Openbox, it's one of the oldest projects featured in this *Roundup*. You can use various options to configure different aspects of the desktop. While most settings apply to the logged-in user, there are several that help influence the desktop as a whole. Similarly, you can use the Moksha Settings Panel to add more bling to the desktop. You can change the theme for the desktop and even for the apps. All the elements have various other tweakable settings as well.



LXQt's configuration manager boasts several options geared towards more advanced users.

Besides the usual settings, there are also several advanced ones. Of note among these are the Power Management settings that help extend battery life by enabling you to adjust the time for deferring various power-intensive tasks.

VERDICT

EQUINOX DESKTOP	**7/10**	**MOKSHA DESKTOP**	**8/10**
LUMINA	**8/10**	**OPENBOX**	**7/10**
LXQT	**8/10**		

You can mould each of these desktops as per your needs.

Lightweight desktops
The Verdict

Taking app launch times as an indication, all the options in this *Roundup* are equally lightweight and responsive, and don't give us much to choose between in terms of performance. So instead we'll concentrate on their other aspects and determine our winner using the tried-and-tested mechanism of elimination.

Lumina loses out for the simple reason that it wasn't designed for Linux. It also isn't available in the repos of many distributions, which makes installation a tall order for most users.

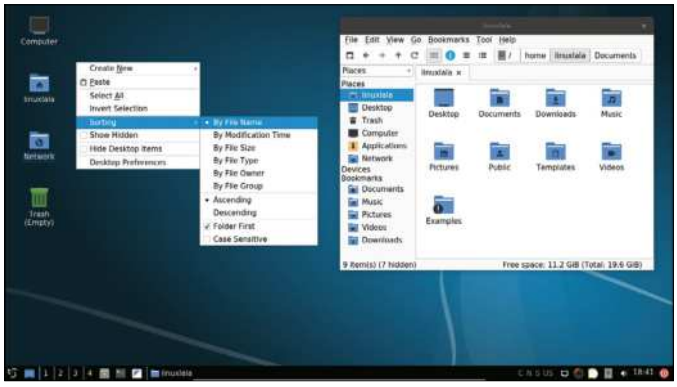
We are really pained to deny Equinox a podium finish, especially since it performs extremely well and doesn't score too badly in terms of usability. However, the project isn't as actively developed as its peers, and you can't always find it in your favourite disto.

Moksha gets on the podium for its beautiful Enlightenment pedigree. It not only looks good but is very responsive and light on resources. As with most desktops designed for a specific distribution, though, Moksha doesn't always sit pretty across other distributions.

Both LXQt and Openbox are mature projects and have been around for several years now. Openbox is surely the faster of the two, but what prevents it from claiming victory is its weak usability. Since it's a bare-bones window manager, users will have to spend some time assembling various components and writing a custom configuration file to create a usable desktop environment. While that doesn't take as much effort as it sounds, it's still an exercise and takes longer than a ready-made solution like LXQt.

LXQt is the result of the merger of the LXDE and Razor-Qt desktops. It boots very quickly and makes judicious use of available resources. Despite consuming a fraction of the resources of its mainstream rivals, LXQt looks good thanks to its Qt underpinnings, which are also the reason for its slightly higher resource use when compared to Openbox.

That said, LXQt will feel at home on a modern machine, but is still light enough to push an out-of-commission computer back into active duty.



1st LXQt 8/10

Web: https://lxqt.org Licence: GPL, LGPL
Version: 0.14.1
A lightweight desktop which still manages to look good.

2nd Openbox 7/10

Web: http://openbox.org/wiki/Main_Page Licence: GPL v2
Version: 3.6.1
The best option for tweakers who love spending time setting up their desktop.

3rd Moksha Desktop 7/10

Web: https://www.bodhilinux.com/moksha-desktop
Licence: BSD Licence Version: 0.3.0
A wonderful option for users who love Enlightenment's bling.

4th Equinox Desktop Environment 6/10

Web: https://edeproject.org Licence: GPL, LGPL
Version: 2.1
A good bet for anyone who can find it in the repositories of their distribution.

5th Lumina 6/10

Web: https://lumina-desktop.org Licence: BSD 2.0
Version: 1.4.0
Optimised for BSD, it doesn't adhere to the sensibilities of a Linux distro.

» ALSO_CONSIDER

Unlike some of other *Roundups*, there's no dearth of lightweight options besides the ones featured here. There's the Sugar desktop environment that is developed as an educational desktop for the OLPC project. Although not a regular desktop environment, it's a fantastic option if you are looking for an environment custom-built for an educational computer.

One of the most popular options for cutting the flab from desktop environments is to simply run their lightweight window managers instead. We've covered some in the *Roundup* and there are plenty of others as well. There are stacking window managers like Fluxbox, Flwm, FVWM, IceWM and JWM that will all run without stressing the available resources of your computer.

Meanwhile, command-line warriors will appreciate the hackability of tiling window managers such as Qtile, i3 and xmonad. Many of them are available in the repositories of popular desktop distributions. However, they score poorly in terms of usability and for that reason aren't everybody's cup of tea. LXF

Roundup

AIO System Rescue Toolkit » ALT Linux Rescue » MorpheusArch » SystemRescueCd » Ultimate Boot CD

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

Rescue distros

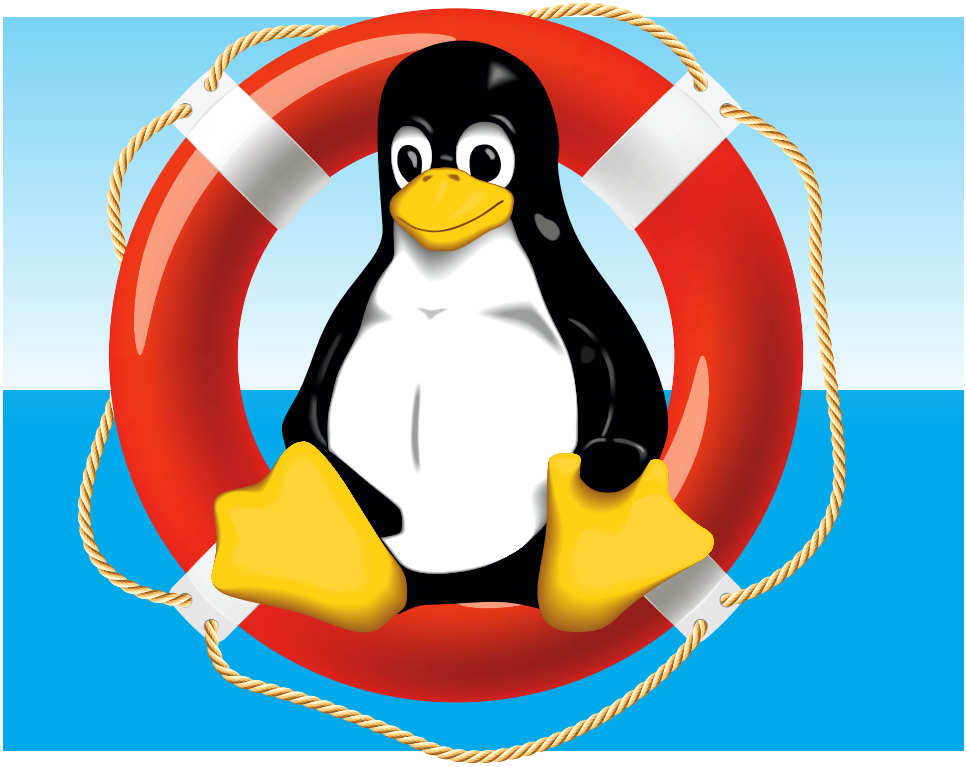
Linux is pretty resilient, but when things get out of hand, **Shashank Sharma** uses one of these specialised distros to rescue an installation.

HOW WE TESTED...

Our parameters for testing haven't really changed from the previous rescue *Roundup* in **LXF209**: we're still looking for the distros with the widest collection of tools that'll help us with all kinds of repair and rescue operations.

However, that's not the only parameter we're interested in and it really isn't even the most important one. Instead of the number of tools, the effectiveness of the complete distro gets more weighting in our books. Distros that help you wade through the multitude of tools and home in on the right tool for the job are rated higher than distros that only claim to bundle the largest number of rescue utilities.

Since many of these utilities interact directly with disks and the crucial data they house, they can do more harm than good if used improperly. For a specialised task such as this, documentation thus plays a very crucial role.



Linux has no shortage of healing utilities and tools that are designed to put broken computers back into active duty. Sure, the Linux kernel by design (and Linux distros by extension) has a higher threshold for breaking down, but a hardware failure or a clumsy operator can still wreak considerable havoc.

The plethora of repair tools help users get out of all kinds of sticky situations. However, due to the sheer number of options on offer, the average Linux desktop user might not always choose the best tool for the job. Also, while you can find the tools in the official repositories of virtually all mainstream distros, since many of them are CLI-based they do not show up in application menus – so many greenhorns are actually oblivious of their existence. This is where the specialised rescue distros come into the picture. Instead of simply collating a vast number of healing tools, these distro help users find the right tool for the job. We've looked at repair and rescue distros in the past (see **LXF209**, page 24). But there's a pretty high rate of attrition for these specialised spins, so much so that besides two, all other candidates in the previous *Roundup* are now defunct – including the winner, Rescatux, so a rerun is more than overdue...

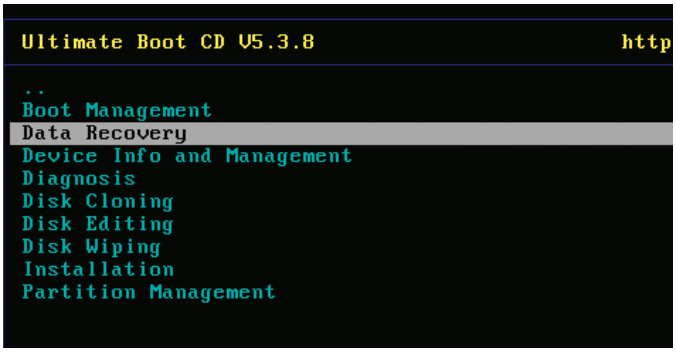
The range of tools on offer

What's in the goody bag?

Should you be looking for a wide range of repair tools and utilities, you'll be disappointed with MorpheusArch. The distro has a very focused approach and only really includes *TestDisk*, along with its companion *PhotoRec* utility, for recovering files. The tools support all popular file systems and can identify files in over 400 formats. In addition to files, you can also use them to find lost partitions, rebuild the boot sector, fix FAT tables, locate the backup superblock and more. Experienced users can use its Arch base to fetch other rescue tools.

Another Arch-based rescue distro, SystemRescueCd is a slightly better option. Unlike MorepheusArch, this one ships with several repair and rescue tools and utilities, and supports all the important filesystems. There are multiple recovery tools, filesystem-specific repair tools, and a couple of boot images including HDT to query the hardware. There's also *FSArchiver*, which can help save and restore a partition table, or the contents of a filesystem, to a compressed archive file.

Then there's AIO System Rescue Toolkit, which readily admits to not shipping with the entire gamut of rescue tools on offer. It instead includes only the most useful ones that its creator, a skilled technician, has used out in the field to fix machines. Besides tools like *Boot Repair*, *Clonezilla* and *TestDisk*, there are also a couple of stress-testing and benchmarking tools. The primary goal of the distro, however, is to fix Windows machines, and it has the largest collection of tools for this purpose as compared to the other options in this *Roundup*.



Navigate through UBCD's diverse menu to find the tool you're interested in, and boot straight into its streamlined environment.

Stress-testing and benchmarking utilities are also included in ALT Linux Rescue, which packs in over a hundred CLI tools, including several versatile ones. The distro also has several utilities to help recover deleted files and fix common issues with both Linux and Windows installations.

Ultimate Boot CD (UBCD) leads the pack with the largest and most diverse collection of utilities. You can use the distro to query and stress-test hardware and peripherals connected to a system. It can be used to restore BIOS settings as well as to restore and backup the CMOS. UBCD enables you to fix bootloaders, recover lost passwords and deleted files and will also help you tweak the Windows Registry without booting into the installation.

VERDICT

AIO SYSTEM RESCUE TOOLKIT	8/10	SYSTEMRESCUECD	6/10
ALT LINUX RESCUE	9/10	ULTIMATE BOOT CD	9/10
MORPHEUSARCH	3/10		

With the exception of MorpheusArch, the others can get fair bit of mileage with the other rescue distros in the Roundup.

Customisation

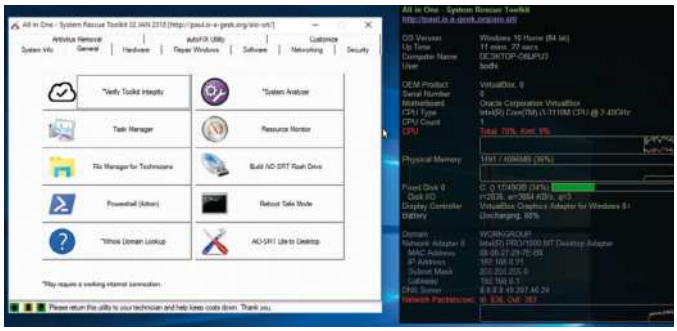
Do they help ease admin access?

SystemRescueCd doesn't offer much in terms of a customised interface. The distro has a simple boot option that takes you to its Xfce desktop, which doesn't offer any conveniences or customised interfaces to help orientate new and inexperienced users.

The MorpheusArch project does produce an *ncurses*-based script called *LinDiag* to assist with common system admin and recovery tasks. You can use it to check services, diagnose the network, manage packages and recover data from drives. However, the script oddly isn't included in the distro by default, and you'll have to clone it from GitHub.

ALT Linux Rescue also doesn't offer much in terms of customisation. One of the few aids it does have is the forensic-mode boot option that brings up the distro without activating the swap or mounting any partition. There are also a couple of scripts that automatically mount attached disks and fix any issues with the master boot record, which is rather convenient.

In contrast, AIO System Rescue Toolkit has a bunch of CLI tools, but to make for easier access it has a shortcut on the desktop for all of them. Besides the Live environment, you can also boot the distro's CD/USB inside Windows to access all its Windows-specific repair and rescue tools via a custom interface.



If you manage a bunch of Windows machines, AIO System Rescue Kit can help you to keep them in perfect condition.

What helps UBCD standout from its competition is its custom boot menu. The distro ships with no graphical desktop and instead boots to a very well laid-out menu. The customised text-based menus help guide you to the relevant utility for your issue. The menu is logically arranged and individual entries have useful information to help identify them.

VERDICT

AIO SYSTEM RESCUE TOOLKIT	8/10	SYSTEMRESCUECD	3/10
ALT LINUX RESCUE	7/10	ULTIMATE BOOT CD	9/10
MORPHEUSARCH	5/10		

New users will appreciate UBCD's customised boot menu in particular.

Usability

What's their user experience like?

When it comes to distros that package specialised tools, it takes a lot more than the sum of its parts to win us over. Stuffing distros will loads of tools will help experienced users, but won't be of much help in the hands of the average desktop user looking to rescue an unbootable machine.

It's because of this that usability becomes such a critical factor for specialised distros such as the ones in this *Roundup*. Also, remember that there's more to usability than just beautiful interfaces; having relevant information on using the tools at hand is just as essential as picking the right tool for the job.

Finally, just because a distro is usable irrespective of a user's level of experience doesn't mean it has less powerful tools than a distro designed for experienced users.

System Rescue Toolkit 8/10

The distro is unique in that it has a couple of user modes. You can transfer the ISO to a CD or a USB and use it as a regular Live disk, which boots into a tweaked LXDE desktop. The desktop has icons for all the apps including command-line versions, which is really helpful for first-time users.

Besides the Live environment, you can also run the distro from inside Windows. In this case it fires up a custom repair utility that lists all the available tools in its repository to help you repair a broken Windows installation.

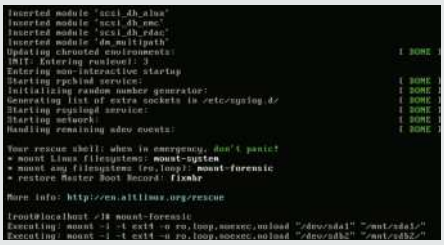
The custom app has multiple tabs, each of which houses apps for a particular area such as Hardware, Software, Networking, and such. While the custom app is intuitive to operate for anyone who is familiar with troubleshooting, there's also a very helpful *AutoFix* utility which can help novices repair many common issues with a Windows installation.

ALT Linux Rescue 5/10

Unlike the other distros in the Roundup, ALT Linux Rescue boots straight to the shell and is missing a graphical desktop altogether. This clearly points to the fact that it's designed for experienced users who are not only familiar with the bundled tools but also know the ones they need to use to fix the issues affecting a system.

Besides the couple of scripts mentioned earlier, you get no other conveniences. In addition to the regular boot option, the distro has a couple of others. There's the aforementioned Forensic mode, which minimises interactions with the computer's disk to help you conduct forensic investigation on the system.

The other boot option is designed to rescue headless machines. It automatically assigns an IP address to the machine, generates a random root password and starts a SSH server to help you access the machine remotely.



Help and support

For when you need some hand-holding.

MorpheusArch is one of the two distros in this *Roundup* that doesn't have any sort of help information of its own. You do get a brief illustrated guide on *LinDiag*'s GitHub page, but there are no forum boards or mailing lists on which to seek help, so this distro is only designed for people who want to use the bundled recovery tools on an Arch base.

ALT Linux Rescue isn't any better in that it only gets a single page on the ALT Linux website, which lists the distro's capabilities and some of the marquee utilities. There are Russian forums and mailing lists for the main ALT Linux distribution, but the Rescue edition doesn't have any official support avenues.

In sharp contrast, AIO System Rescue Toolkit has as a detailed FAQ with a list of all the utilities and a couple of videos to help you find your way around the distro. There are no forum boards, but the developer has created a page on the website where users can post questions.

SystemRescueCd's website hosts a quick-start guide as well as detailed instructions on basic and advanced use. There are also instructions for experienced campaigners, such as the guide on how to make a custom version and backing up data from an unbootable Windows computer.

With a tutorial section wiki and a detailed FAQ designed to familiarise users with the distro, UBCD surpasses its competition by a long way in this regard. The tutorial section, however, is just a pointer to a long list of user-contributed tutorials curated from the forum boards, so it could still be a little more friendly. The wiki has a list of all the tools in the distro, along with useful notes on their capabilities.

VERDICT

AIO SYSTEM RESCUE TOOLKIT	7/10	SYSTEMRESCUECD	9/10
ALT LINUX RESCUE	5/10	ULTIMATE BOOT CD	9/10
MORPHEUSARCH	2/10		

SystemRescueCd and UBCD are the only distros that actively resolve queries on forum boards.

MorpheusArch 3/10

MorpheusArch has fairly limited appeal as compared to some of the other repair and rescue distros in the *Roundup*. It only really advertises the inclusion of *TestDisk* and *PhotoRec* utilities that help users to recover accidentally deleted files. The distro boots up to a locked LXQt desktop and there's no mention of the default password ('arch') on the project's website, which is a serious oversight.

The LXQt desktop has a handful of apps including *Firefox*, *Zenmap* and *LFTP*. The desktop provides no indication on how to proceed, which is rather disorientating. The lack of getting started documentation inside the distro or even on the project's website doesn't help matters either.

One of the best things about the MorpheusArch project is the assistive *LinDiag* utility. However, it isn't included with the distro and you'll have to manually pull it in, which again is not made clear.

SystemRescueCd 7/10

Earlier versions of the distro had a comprehensive boot menu with over a dozen boot options. These have been dropped in the latest Arch-based release, which only has two.

There's an option to copy the entire system to RAM and the default option will boot the system to a console, which lists the command to configure the keyboard layout and start the graphical environment. The distro uses the Xfce desktop and houses a handful of essential productivity apps in its menus including a web browser, a VNC viewer, a password manager and some developer-friendly tools.

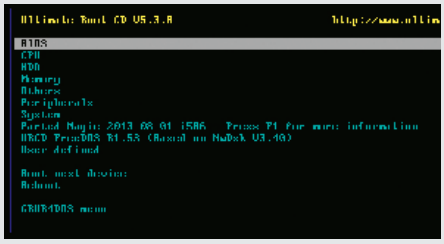
A majority of the rescue tools in the distro are CLI-based and just like MorpheusArch, the distro doesn't give any pointers to help users get started. Its saving grace is its website, which is flush with all sorts of detailed documentation to help first-time users use the distro.

Ultimate Boot CD 9/10

Unlike the other distros here, UBCD is unique in that it directly boots into its customised text-based menu. The menu divides the packaged utilities into categories based on the part of the computer that they influence, such as HDD, Peripherals, System and such.

Step into the category to view the list of entries for the individual tools. Some categories, such as HDD, are further divided into different tasks, such as Boot Management, Data Recovery, Disc Cloning and Disk Wiping, and more.

As you move across each tool in a category, the menu displays brief but useful information about it. The menus might be text-based, but the clear categories and helpful descriptions mean the distro can be easily navigated and used by inexperienced users. Coupled with the useful information on the project's website, UBCD makes for a great choice.



Security and privacy features

What can they do besides repair and rescue?

While the primary purpose of these distros is to rescue a broken computer, many also pack in a bunch of tools to help secure your system and prevent privacy leaks. That's because it isn't uncommon for security issues to bring down a computer. On occasion, you might also have to sanitise an installation before bringing it back up in order to ensure the health of the other machines it interacts with on the network.

MorpheusArch has the *nmap* tool for exploring the network, which you can use through the *LinDiag* script. The distro also has a couple of utilities which can wipe filesystem signatures from a device and also set hard disk parameters.

AIO System Rescue Toolkit also includes *nwipe* to help securely zap files. ALT Linux Rescue includes tools to securely delete files and investigate security breaches. The distro is designed for several scenarios, but unfortunately ensuring privacy isn't one of them.

SystemRescueCd does take steps to enhance your privacy if you're using a YubiKey. It includes the *YubiKey Manager* app as well as the *YubiKey Personalization Tool* that can help you reprogram the configuration slot on the device to enable additional authentication functionalities such as OTP. Besides these, you'll find several tools to help you delete data securely including *shred*, *wipe* and the *THC-Secure Deletion* tool.

Much like its peers, although ensuring security and privacy aren't among the goals of UBCD, the distro does have a specific category where it lists about a dozen tools, including several low-level utilities, for securely wiping disks. There are also scanners to sniff out viruses and malware.

VERDICT

AIO SYSTEM RESCUE TOOLKIT	4/10
ALT LINUX RESCUE	6/10
MORPHEUSARCH	5/10
SYSTEMRESCUECD	7/10
ULTIMATE BOOT CD	7/10

You can use any of these distros to securely delete files so that they aren't recoverable. Otherwise, their security features vary quite a bit.

Healing capabilities

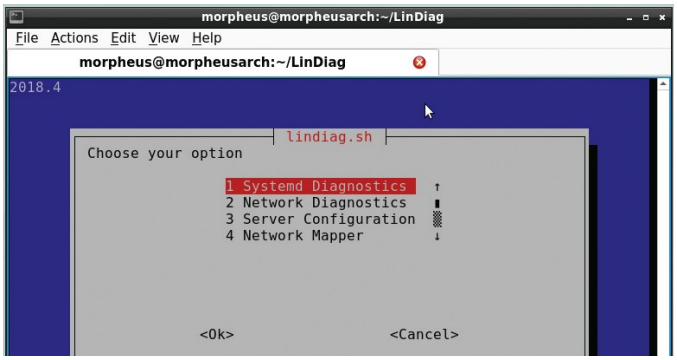
What wrongs can they right?

All these distros will help you wiggle out a tricky situation, irrespective of the ailments that plague your installation. Many of them can also image and clone disks, identify and list damaged sectors, and extract recoverable data from physically damaged disks. However, some of them are more useful than others.

For instance, data recovery is the sole advertised purpose of MorpheusArch, and you can use the distro for little else. AIO System Rescue Toolkit has a handful of rescue tools to fix boot errors and rescue accidentally deleted files. The majority of its tools, though, are for fixing Windows installations, and can be used to fix all kinds of issues with the proprietary OS.

The tools bundled with ALT Linux Rescue can do a wide variety of repair jobs. In the hands of an expert, the distro can fix bootloaders, optimise filesystems, recover accidentally deleted files and partitions, and even diagnose network problems.

Similarly, you can use SystemRescueCd to rectify all sorts of issues with hard disks, including restoring the partition table. It can also save the contents of the filesystem to a compressed archive, securely delete files and recover accidentally deleted ones. You can also use the distro to rescue data from an unbootable Windows machine. As in most other cases, UBCD goes further than the other contenders in that it can fix bootloaders but can also restore BIOS settings as well as restore and backup the CMOS. This is in addition to the usual rescue tasks such as recovering passwords and deleted files. The distro also features several diagnostic utilities.



First-time users can use the LinDiag utility in MorpheusArch to attempt data recovery using the ddrescue utility.

VERDICT

AIO SYSTEM RESCUE TOOLKIT	8/10	SYSTEMRESCUECD	8/10
ALT LINUX RESCUE	8/10	ULTIMATE BOOT CD	9/10
MORPHEUSARCH	3/10		

MorpheusArch is the only distro in the Roundup that has a very focussed rescue objective.

Customisation

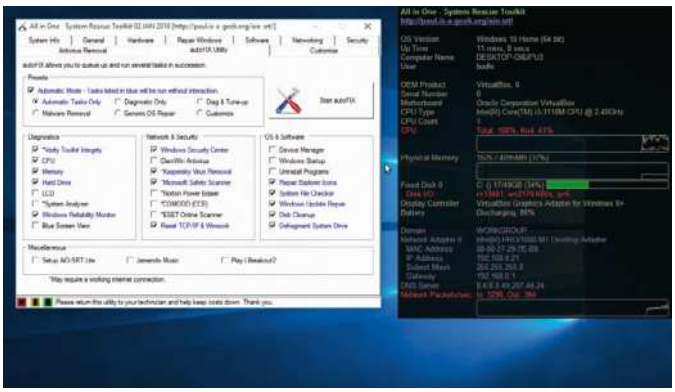
Can you adapt them as needed?

Despite their fairly large collection of apps, these distros don't include *all* the repair and rescue tools available. This is partly because some popular rescue tools have licence restrictions which prevent developers from including them in their distros. In any case, instead of lugging around multiple distros to accomplish different tasks, advanced users and technicians will prefer to flesh out their favourite distro to include all the tools they need.

If you're looking to decompile and assemble your own rescue distro, you won't get far with ALT Linux Rescue, MorpheusArch or AIO System Rescue Toolkit. None of them offers the option to customise the distro, although you can use their respective package managers to pull in additional utilities and flesh out the live environment as need. In addition, AIO System Rescue Toolkit has a Lite version that is designed to give inexperienced users an opportunity to easily fix their installation.

On the other hand, one of the marquee features of SystemRescueCd is its capability to enable anyone to compile customised versions. There were a bunch of scripts to ease the process of remastering older versions of the distro. That process has now been simplified in the newer Arch-based version, in that it can be simply rebuilt using the sources posted in the project's Git repository. You will need to have an Arch installation to be able to run SystemRescueCd's build scripts, which are based on the *archiso* tool.

UBCD's customisation is on a par with SystemRescueCd. The project's website hosts a guide which details the procedure for extracting the ISO image and then customising the environment, by adding your own floppy and ISO images as well as FreeDOS-based apps, before compiling a new ISO image. The website documentation also lists the process for updating virus definitions. One particular benefit of UBCD's customisation mechanism is that since all the modifications are housed within a single folder, you can easily move to a newer version of UBCD without losing any of the changes you've made to the distro.



The Lite version of AIO System Rescue Kit runs through several repair tools to help fix most common issues with a Windows installation.

VERDICT

AIO SYSTEM RESCUE TOOLKIT	5/10	SYSTEMRESCUECD	8/10
ALT LINUX RESCUE	1/10	ULTIMATE BOOT CD	8/10
MORPHEUSARCH	1/10		

SystemRescueCD and UBCD are the only distros that enable you to make custom versions with your own tools and utilities.

Rescue distros
The Verdict

The first step to rescuing a damaged computer is to have access to the right utility, which is why we looked favourably at distros with a large collection of tools. But just having the largest collection of tools isn't of much use in the hands of inexperienced users. Taking that into consideration, the winner of this *Roundup* has to be a distro which can be used by a large number of people, irrespective of skill and experience.

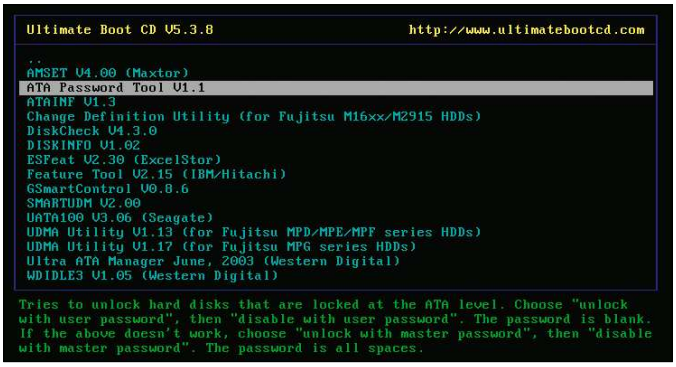
This is why ALT Linux Rescue loses out. While the distro is equipped to handle all sorts of repair and rescue jobs, it won't be of much use in the hands of new users – it's firmly designed for experienced technicians.

MorpheusArch Linux loses out for similar but different reasons. Unlike ALT, the distro boots into a graphical desktop, but all the tools it packs are CLI-based and the distro provides no assistance to help guide users to the right tool for the job. We do like its *LinDiag* utility, which is a nice attempt to make some of the tools more palatable to first-time users, despite its limited area of reach. But the fact that it isn't included by default in the distro is a huge downer.

SystemRescueCd has some of the same problems. Unlike earlier versions of the distro, which had a comprehensive boot menu, the current version is only useful in the hands of someone who is familiar with its cache of tools.

The two options that impressed us the most are AIO System Rescue Kit and UBCD. Both are loaded to the brim with tools and utilities, and have customised interfaces to help you find your way to the right tool. However, AIO loses out to UBCD for its strong focus on Windows-specific repairs. Even its customised menu comes up when the distro runs from inside Windows.

UBCD wins this *Roundup* because it scores highly on both our criteria. The distro is loaded with tools and is still intuitive enough to be of use in the hands of first-timers. Its logically arranged boot menu is also its USP, offering useful information to help you identify the right tool and then boot straight into it.



1st Ultimate Boot CD 8/10

Web: www.ultimatebootcd.com Licence: Various Version: 5.3.8
UBCD packs in enough utilities to justify its name and exposes them via a user-friendly interface.

2nd AIO System Rescue Toolkit 6/10

Web: https://paul.is-a-geek.org/aio-srt Licence: Various Version: 2018-01-02
The most comprehensive solution for rescuing... Windows installations.

3rd SystemRescueCd 5/10

Web: www.system-rescue-cd.org Licence: GPL v2 Version: 6.0.2
Can help you wriggle out of many common issues, if you have the time to read through its documentation.

4th ALT Linux Rescue 5/10

Web: ttps://en.altlinux.org/Rescue Licence: Various Version: 20190306
A good option for experienced users who know the bundled utilities.

5th MorpheusArch 3/10

Web: www.morpheusarch.co.uk Licence: GPL v2 Version: 2018.4
Weirdly, the best thing about the project (*LinDiag*) isn't included in the distro by default, too limited at the moment and too hard to use.

» ALSO_CONSIDER

As we mentioned earlier, many of the rescue distros that we've featured in past issues are no longer maintained. While they might not support the latest hardware, nor have the latest version of the bundled utilities, you can still use them to repair older machines, however. For example there's Rescatux (*Tux with fingers should not be!–Ed*) that has an interesting helper app to help you repair different parts of your installation without needing to be aware of the underlying CLI tool.

Then there's Trinity Rescue Kit, which features an elaborate boot menu designed to help you find the right repair tool. Many repair and rescue tools also roll their utilities into Live CDs that you can use if you know the tool you need to use. For example, *Boot-Repair* has a Live CD which is ideal for fixing all sorts of bootloader-related issues. *GParted* also has a Live CD of its own, which includes several other repair and rescue utilities such as *TestDisk*, *FSArchiver* and more.

Pop!_OS

Let eyecatching-distro aficionado **Jonni Bidwell** show you the ropes of System76's bespoke Linux flavour.

There have been a few notable efforts to humanise Linux. Today we'd say – and feel free to disagree – that Linux Mint and elementary OS are the most friendly-to-use and all-purpose distros out there. These are of course indebted to Ubuntu for providing a solid foundation and superlative package selection – which in turn owes something to Debian. But it's also arguable that much of this popularity is, or at least was, a result of dissatisfaction with Ubuntu's desktop.

People who didn't like Unity loved Cinnamon, and the people who did like Unity didn't like GNOME 3 (which Ubuntu has used since 17.10). Pantheon, the macOS-like desktop of elementary OS, is attractive not just to fans of fruit-based fashion companies, but to anyone frustrated with over complicated configuration, inconsistently styled applications and ugly fonts. Of course, Windows 10 drives a steady trickle of users to Linux too, as the Start menu begins to resemble some sort of ever growing billboard farm, and updates constantly get in your way.

Hardware compatibility is an important concern too. Today, distros have to cater to all kinds of new-fangled configurations: HiDPI fractional scaling, multi-monitor and hybrid GPU setups, disk encryption. Users want easy access to the latest software, and developers want easy access to their preferred development tools. Providing such features is especially important if you're a manufacturer of Linux systems like Colorado-based System76, and that is why it developed the mysteriously punctuated Pop!_OS. Let's take a look at it.

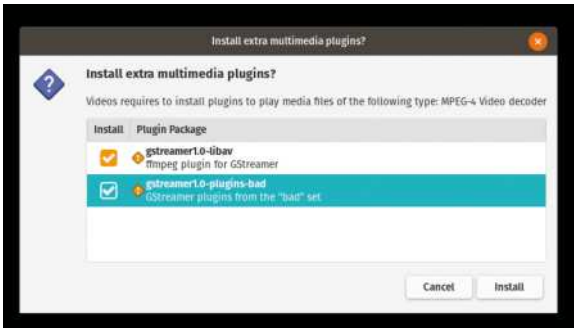
Hopefully by now you'll have had a play with Pop! and have seen quite what all the fuss is about. You may even be itching to install it, in which case check the step-by-step guide. Before you do though, you should be aware that if you have Nvidia hardware and generally want life to be easier, you should definitely use the Nvidia edition of Pop from https://system76.com/pop. This comes with Nvidia's proprietary driver set up nicely out of the box, which is very useful.

The AMD/Intel edition contains only open source drivers, including the Nouveau driver for Nvidia cards, but that's not much use for AAA gaming or CUDA programming. If you have any problems booting a live disc or booting a Pop!_OS USB, ensure that you have disabled secure boot in the UEFI settings (often found by pressing F2 or Delete at boot time). Different BIOSes use different hotkeys to invoke a boot menu, but F11 and F12 are popular. Like most unfamiliar distros, you may find a few quirks when you install that you'll need to work out, but a little resourcefulness and checking the online FAQs should minimise this.

Pop it like it's hot

The guide covers installing alongside Windows; we'd always recommend installing on a different drive if you have one available though. UEFI booting means Windows usually manages to keep itself to itself, or keep its breaking to itself. If you use classic BIOS you might find Windows blithely overwriting the bootloader from time to time, which is annoying, but can be remedied. Either way, it's important to disable Fast Startup in Power Options in the Windows control panel, otherwise you won't be able to boot another OS. If you're setting up Pop as your only OS, you can safely ignore all the partitioning instructions in the walkthrough; just select 'Erase entire disk' from the installer and everything will be set up for you.

You should also be able to safely use the Pop!_OS Installer's partitioning tool to resize the NTFS partition, but it's probably safer to let the devil (Windows) do the devil's work (rearranging NTFS structures). If your EFI partition is less than 512MB – some Windows installs seem to settle for 100MB – then you'll need to use a third-party tool to resize this and possibly move the Windows System reserved partition, which we can't possibly recommend. Hopefully this isn't you and you have no need to worry. *Distint* (Pop's bespoke and delightfully simple installer) will hold your trembling



The Videos app offers to install any codecs you might need, or alternatively install VLC or MPV.

hand throughout the installation process.

Pop!_OS doesn't really resemble Windows or macOS, and it only vaguely resembles Ubuntu's current Gnome setup. But it is easy to get to grips with; hit the Super (Windows) key or click the top left corner to access the activities view, then start typing to find installed applications. That will also show you applications you can install from the *Pop!_Shop* (okay, that does have a nice ring to it) as well as any recently opened matching files. Our initial install weighed in at around 5.5GB, which is pretty modest by today's standards. As with many distributions *LibreOffice* comes bundled, but beyond *Firefox* and core tools there are no crazy-large application inclusions. There are a few gems included, such as the delightful *Geary* email program and System76's own *Popsicle* USB-writing tool. There's also *Eddy*, which like its rhyming brethren *Gdebi* is a GUI tool

If you choose to install Pop on its own drive, you can enable full-disk encryption so your data is safe at rest.

» THE LITTLE THINGS

A lot of what distinguishes Pop!_OS is its behind-the-scenes attention to detail. Things that are traditionally annoying are simple with Pop! System76 has written its own HiDPI daemon, which you can read all about at **http://bit.ly/1xf252hidpi**. It makes light work of whatever complicated display arrangement you're running, and can even manage mixed Hi and LoDPI displays with cunning scaling. Wayland is disabled out of the box, but you can enable it by editing **/etc/gdm3/custom.conf** and commenting the line:

```
WaylandEnable=false
```

You'll then see the usual cog icon when you type in your password at login, and you can choose between Wayland and X.org sessions. Wayland in some multi-DPI situations performs better than X, and in our testing we only ran into a few niggles – mostly using Fedora's experimental *Firefox Wayland* flatpak from **https://firefox-flatpak.mojefedora.cz**.

It's easy to dismiss Pop!_OS as just another Ubuntu clone, but we don't do features on insipid distros, so there must be something to it. Beyond what we've written about here, there are so many other little tweaks to make life easier. These have been done cleverly so you hardly notice them. With users at the heart of its design and that design informed from the unique viewpoint of an OEM, Pop is just lush. It's perfect as a beginner's distro, but it hasn't been 'dumbed down' in any way. It's so flexible and powerful that more advanced users will enjoy it too. Heck, we just installed it on the official **LXF** laptop (*Jonni's Eee PC!–Ed*). So there.

for installing DEB files – ideal for those who are command-line averse.

From the Activities view find the *Pop!_Shop* launcher either by typing it or selecting the Show Applications button on the dock to the right, pausing for a second to admire the bold icon design. You could say that Pop!_OS's initial app offering is quite sparse, but the idea is that you add only what you need to it. Not everyone wants *Visual Studio Code* or *Slack*, but if you do they're very easy to find in the *Pop!_Shop*.

Today's application picks are brought to you by the letter G: *Goxel* – making voxel art is surprisingly calming; *Gnome Twitch* – for watching Gaming On Linux videos; and *GIMP*, which is still the best way to manipulate images.

Naturally, command-line installation with *apt* is possible too, and many packages don't have a *Pop!_Shop* entry. One thing missing from the Pop install is the OpenVPN plug-in for NetworkManager, which if you use a VPN (or envisage doing so in the future) you'll want to remedy with:

```
$ sudo apt install network-manager-openvpn-gnome
```

We hope your Pop install doesn't break, but if it does help is at hand in the form of a recovery partition. This essentially lets you boot to something like the Live environment on the DVD, from whence repairs can be carried out. Press Shift and Escape while the system is booting and select the recovery environment from the menu. If that doesn't work you can perform a Refresh Install, which resets all system configuration and applications, but retains your user information and documents, much like the equivalent Windows utility.

Speaking of booting the system, Pop is pretty much unique among major distros in that it uses *Systemd-boot* – formerly *Gummiboot* – to boot the system on UEFI installs. This means that for once we can say you won't run into any horrible *GRUB* problems on this distro, unless you use classic-BIOS booting. You shouldn't run into any *Systemd-boot* troubles either, by the way, as System76 test all kinds of weird and wonderful configurations on its hardware.

The included video player is fine, but you may prefer to install *MPV*, which handles its own codecs. Once installed it also takes advantage of whatever hardware acceleration is on offer (VDPAU, VA-API or some unholy combination of the twain is configured out of the box) to decode video without bothering your CPU.

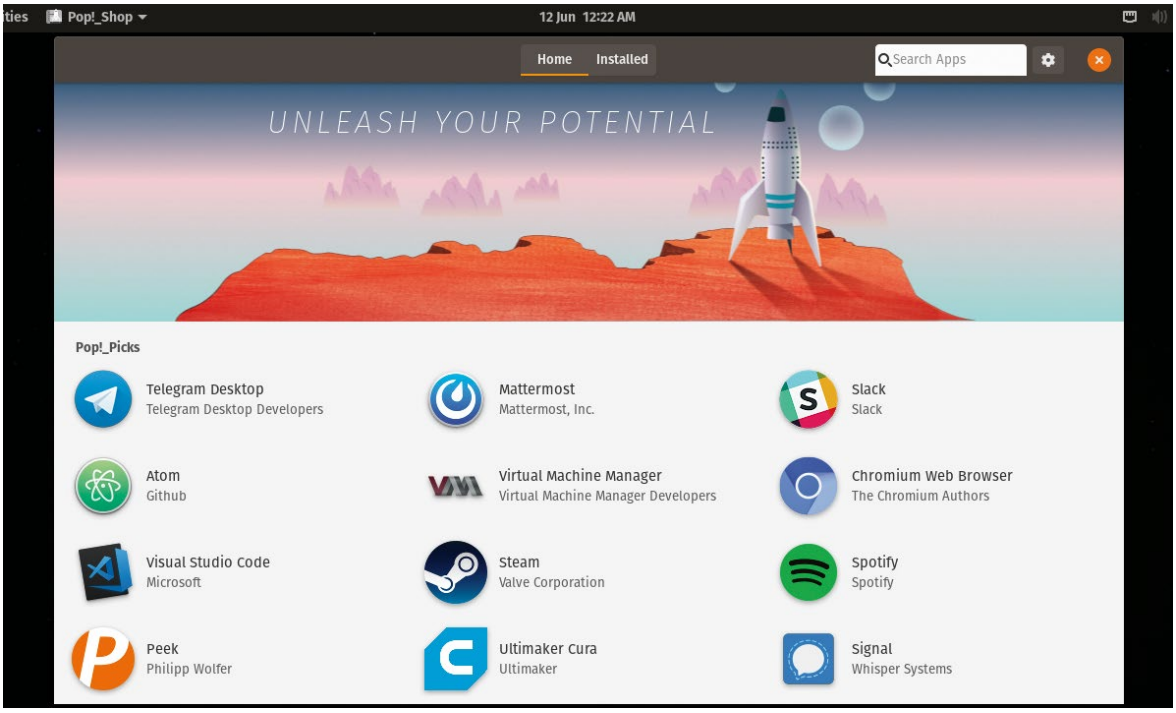
The Pop!_OS team has done a great job of preparing documentation too. There's a growing list of guides to common tasks at **https://support.system76.com/articles**. For example, if all the abstruse references to terminal commands throughout this magazine make no sense to you, find out more in the Terminal Basics article there.

The team are enthusiastic about getting more people involved with development, too. They even mark out specific bugs in their project as being 'bitesize', suitable for aspiring coders to cut their teeth upon. We have to credit them for this, as the growing divide between users and developers will only be lessened if more people get involved, and such outreach efforts are a fine way to encourage that kind of involvement.

We also must credit System76 for its excellent open hardware efforts. Besides contributing to open firmware causes, its flagship Thelio machine includes a custom daughterboard for managing airflow. The specs for this are completely open, and System76 hopes to develop more hardware that follows this mode. We're definitely excited to see more. LXF

CHOOSE YOUR OWN ADVENTURE

"Pop!_OS's initial app offering is quite sparse, but the idea is that you add only what you need to it."

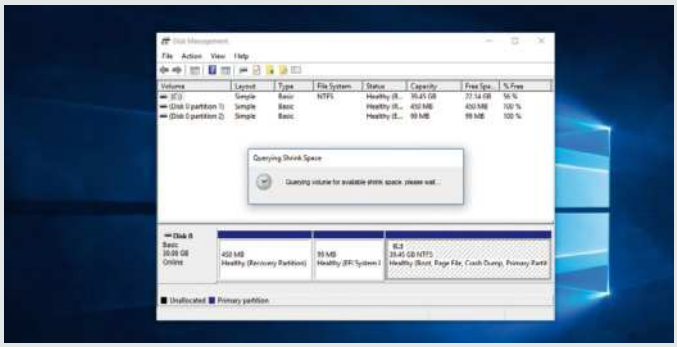


The shopfront offers a good mix of popular proprietary and open source applications.

INSTALLING POP! ALONGSIDE WINDOWS

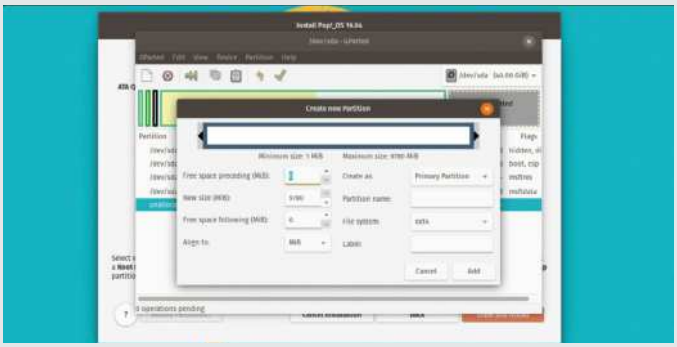
1 Write an install medium

If you have an Nvidia graphics card you'll want to fetch the Nvidia build from **https://system76.com/pop** and burn it to a DVD or write it to a USB stick. If you have trouble booting Pop!_OS from the **LXFDVD** (or have no DVD drive), you should write the ISO file from the **Pop/** directory (or the link above) to a USB.



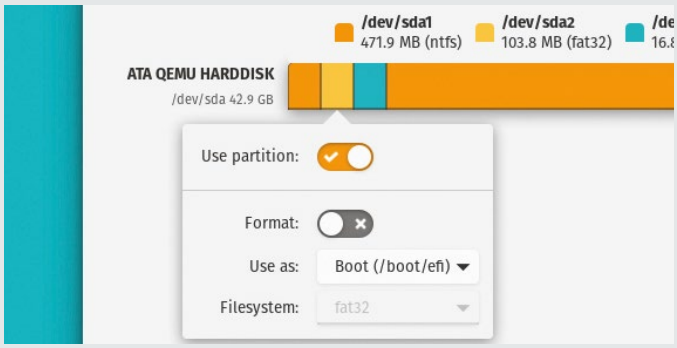
2 Make space

Start windows and open the Disk Management tool. The easiest way to find it is by opening the start menu and typing the first few letters of 'disk man'. Right-click the C: drive and choose Shrink Volume. Make some space for Pop. Pop recommends at least 20GB, but if you can spare it, use it.



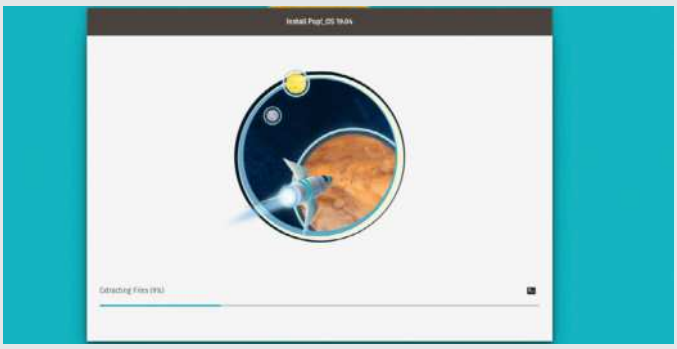
3 Boot Pop!

Boot Pop!_OS and start the installer. After answering some localisation questions choose the Custom (Advanced) option, unless you're happy to get rid of everything on the target drive. Select 'Modify partitions', right-click the Unallocated Space and click New. Go with the defaults and then click Add and finally Apply the changes.



4 Select partitions

Exit the partitioning tool and click the new partition. Click the 'Use partition' switch, again leaving the defaults. Click the EFI partition (it's usually second from the left on a Windows install and is FAT32-formatted) and again use it as a Boot partition, but please don't format it as that will break Windows.



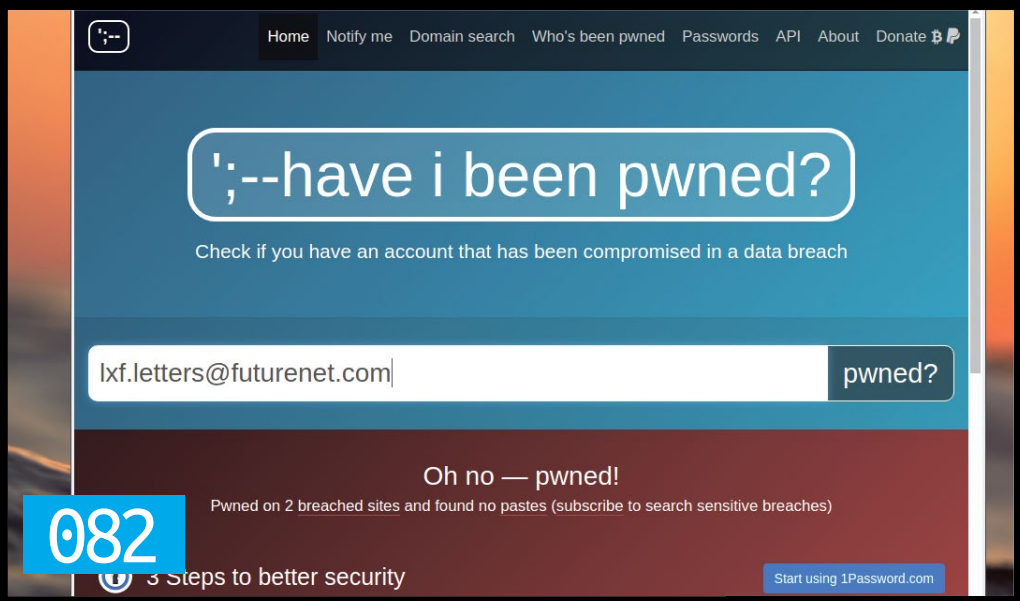
5 Lift off

Hit Erase and Install to begin the fun. Either watch the progress bar intently or make a cup of tea. When the installer's done hit Reboot and do some final setup steps. You can enable location services and connect online services. Finally, set up a user and password, then click the big tick to start using Pop!_OS.

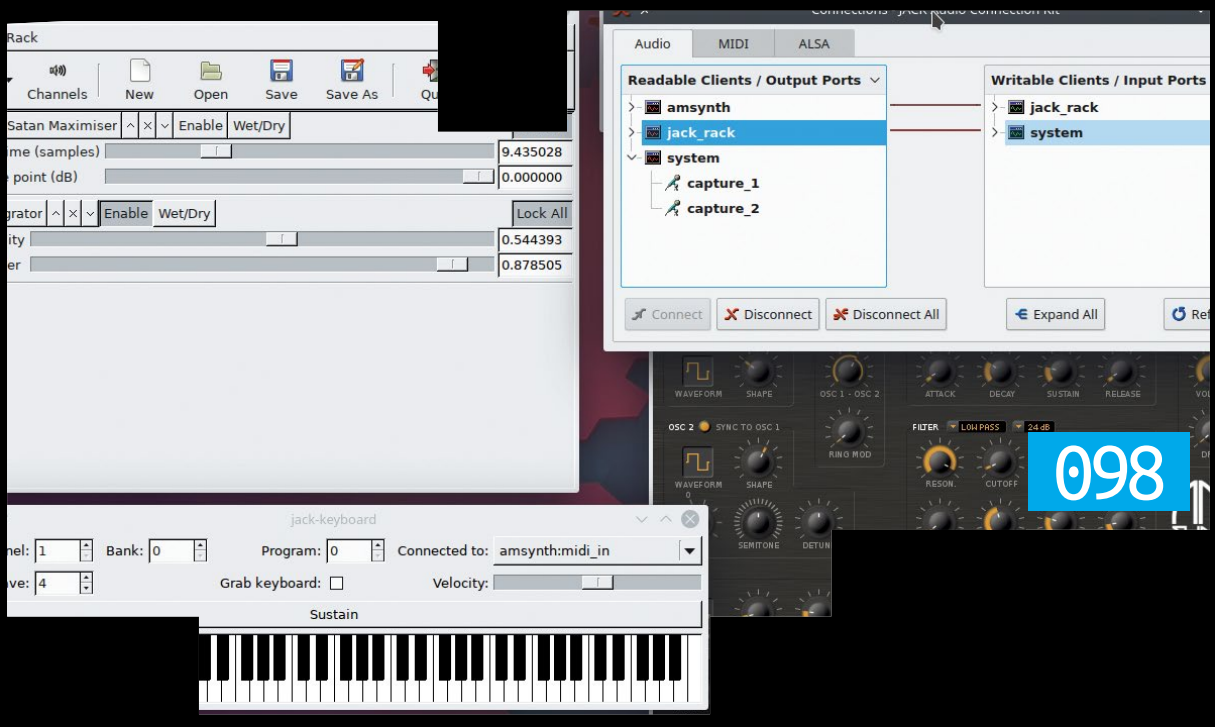
6 Update and augment

Updates will probably be available. You should install these when prompted, then switch from the Installed tab to the Home tab. Here you can peruse the multitude of applications awaiting your attention in the Pop!_Shop. Pop is deliberately lean out of the box, so you may want to install some workhorse applications at this point.

Do more

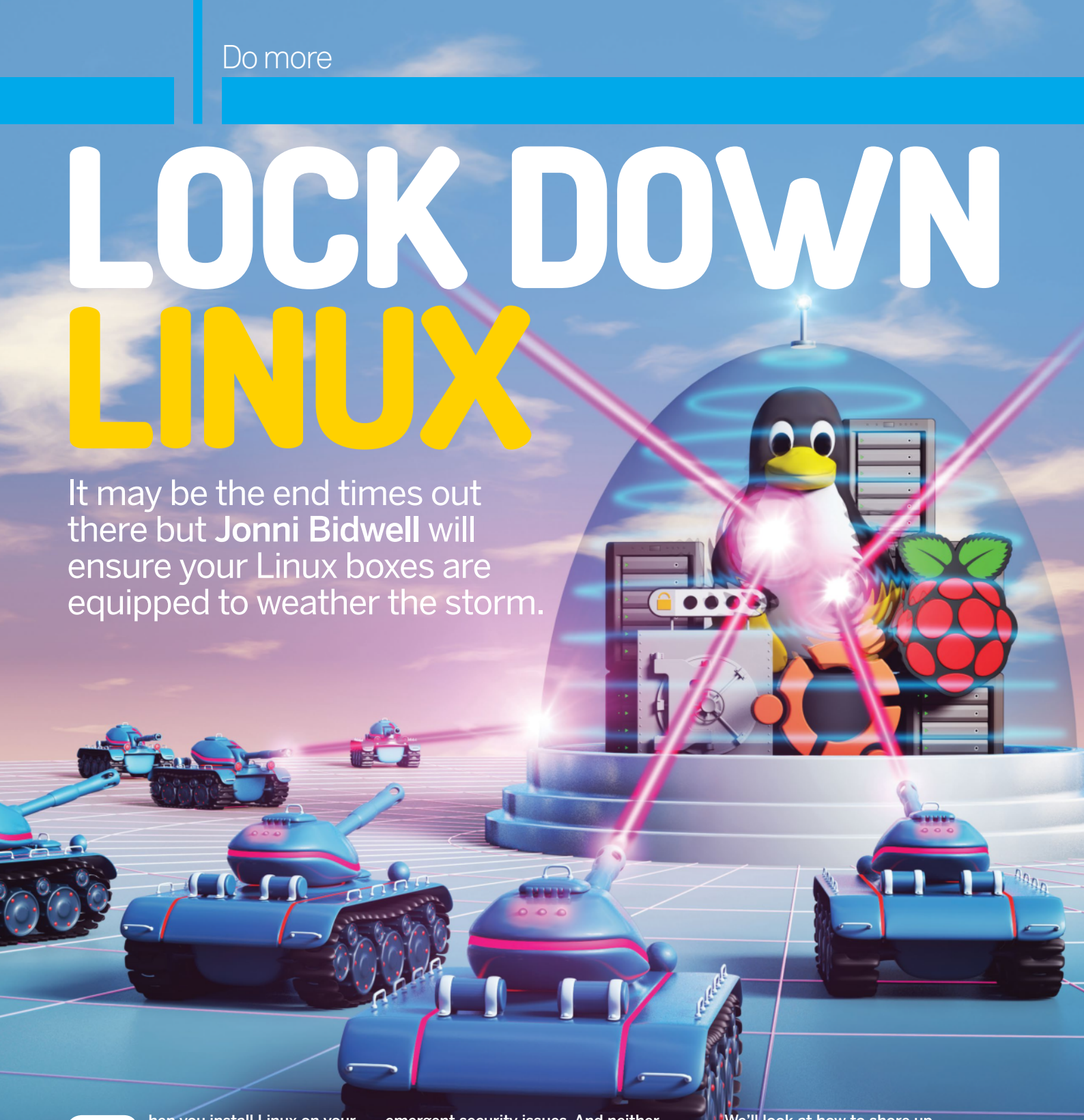



068


Rack
Channels
New
Open
Save
Save As
Satan Maximiser
Enable
Wet/Dry
me (samples)
9.435028
point (dB)
0.000000
Enable
Wet/Dry
Lock All
0.544393
0.878505
jack-keyboard
Bank:
Program:
Connected to:
amsynth:midi_in
Grab keyboard:
Velocity:
Sustain
Audio
MIDI
ALSA
Readable Clients / Output Ports
Writable Clients / Input Ports
amsynth
jack_rack
system
capture_1
capture_2
jack_rack
system
Connect
Disconnect
Disconnect All
Expand All
098

LOCK DOWN LINUX

It may be the end times out there but **Jonni Bidwell** will ensure your Linux boxes are equipped to weather the storm.



When you install Linux on your desktop, then as long as you install an up-to-date distro the chances are you're reasonably secure. The same is true for servers, as long as you choose a strong password (or disable password access altogether and use SSH keys instead). There's a faction of the Linux-using populous that still likes to bang the "Linux is more secure than Windows" drum, but this isn't really true anymore. Both Linux and Windows have multiple layers of security coded by very smart people. Both Linux and Windows rapidly patch emergent security issues. And neither Linux nor Windows can do a whole lot about flaws in whatever software people choose to run on them, and they certainly can't do anything about users configuring that software in an overly permissive manner.

WHAT'S TO COME...

"We will look at everything from next-gen logins with hardware tokens to basics like SSH keys."

We'll look at how to shore up defences on Linux, whether on the desktop, server or up in the clouds. We'll cover passwords, keys, firewalls and much more to keep your data safe. We'll focus more on security than privacy, so won't be talking Tor, VPNs or Whonix, but there's no reason why these can't be used with the setups we'll discuss.

What we will look at is everything from next-gen logins with hardware tokens to basics like SSH keys. We've also got tips for shoring up *Nextcloud* and more. Let's start with a survey of Linux security features, and how they get thwarted.

The state of Linux security

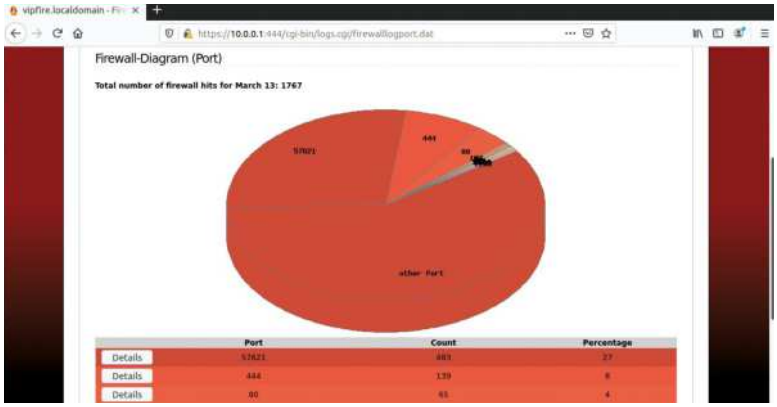
Linux provides more security features than you can shake a stick at – more often than not it's users that are a weak point.

Mainstream Linux distros provide a huge amount of security features out of the box. Some have been around for a long time (address space layout randomisation, having services drop root privileges when they don't need them, mounting removable drives with the noexec option so they can't launch binaries) and some are quite new (Spectre and microdata sampling protections).

A few desktop Linux distributions (Ubuntu, Mint, Solus, Pop!_OS) offer full disk or home directory encryption out of the box too, which we'd highly recommend you do on your laptop, and if you're handling sensitive data it's worth considering on your desktop too. Fedora (and its commercial cousin RHEL) enforces SELinux accounting, which takes permissions and access control lists (ACLs) to a new level, sandboxing apps with fine-grained configurations. AppArmor on Ubuntu does much the same.

Disk encryption is usually done through LUKS and device mapper, and will secure data at rest. However, once the encryption password is entered, that data is (physically) accessible as long as the device remains turned on. With home directory encryption, **$HOME** is usually unlocked until you log out. Similarly, personal data on modern mobile devices is protected by a pin code, pattern or fingerprint. This is why when the FBI or NCA or another three-letter organisation suspects you've been using your devices for no good, they tend to swoop in and grab those off you while you're using them so that they're unlocked. Then a USB dongle is usually fitted, which sends benign keystrokes to ensure the device stays awake and doesn't lock. Usually a power source is connected too, as that would be embarrassing.

If you leave your NAS open to the internet, you better hope it's someone nice like Matthew Garrett who finds it.



We do love pie charts, especially when they tell us no one's trying anything particularly crafty to breach our IPFire defences.

Most new software is installed with a safe, sane configuration, but default usernames and passwords are still common. More often than not users have to tweak initial configurations to suit their requirements, and these tweaks only get as far as making the thing work, not making the thing secure. Where this is most dangerous is running services, because you're pretty much inviting the world to use your machine. You may want the world to see your website, but you don't want to allow them to abuse the machine running it. On the desktop, web browsers and email clients are the main conduits for nasties, and we tend to cover those in our privacy-centric features. Over the page we'll show you how to change your password habits with a hardware key, and how to use *IPFire* to protect your home networks. As for servers we've got all kinds of tips.

» UEFI AND SECURE BOOT

When UEFI was introduced to address the diverse and varied shortcomings of using BIOS to initialise hardware, it was met in some circles with a chilly reception. Most of this centred around UEFI's Secure Boot extension, which at the time made it hard for consumers to replace Windows 8 (which at the time was being shipped on new machines), or even boot a Linux distro. Much of that criticism was unjust, Secure Boot is meant to enable administrators to limit which bootloaders can run and which kernels they can boot.

Most x86 hardware ships with Microsoft's Secure Boot public key pre-baked in, which permits booting only Microsoft-signed bootloaders. Some Linux distros provide a boot manager signed by Microsoft – there are two approaches here, Fedora's *Shim* (also used by Ubuntu and a few others) and the Linux Foundation's *PreLoader*. So these will work fine with Secure Boot, but some other distros will require it to be disabled. If you have full control over Secure Boot though, you can enrol your own signing key into the firmware and allow booting for only those OSes you deign worthy.

If your machine has a TPM chip, you can use this to store, for example, LUKS disk encryption keys. These can be verified against a register in the TPM, so that if another OS is securely booted the disk still won't be unlocked.

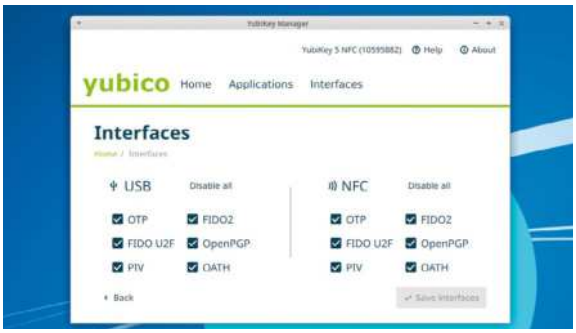
Hardening your desktops

Tighten up your login regimen with hardware tokens, and harden your home directories with the latest Systemd feature.

Human beings aren't really username and password people. Anyone who claims that email is dead would do well to recall that it's often the only way to reset passwords for all those web services we only sign into once in a blue moon. Naturally, we should all be using password managers and that situation should never arise, but to err is human and all that. Still, there are alternatives and augmentations to passwords that provide convenience and security. Major desktops on Linux don't yet provide the face/retina unlock features of Windows 10, but that will change. In the meantime we can use a variety of other means to log into our glorious desktops and online services.

If your laptop and mobile phone have NFC (near-field communication) capability, then there's currently not an awful lot you can do with that on Linux. However, the next edition of *Chromium*, version 81, will introduce Web NFC, which will enable you to authenticate payments or fill in forms with a gentle tap from your phone.

Meanwhile, there are a number of hardware tokens, such as the Nitrokey, YubiKey and Librem Key, that you can use to secure logins to your Linux box and any number of online services, by the magic of time-based one-time passwords (TOTPs) or the FIDO Universal Second Factor (U2F) protocol.



If you run into YubiKey difficulties consider disabling any interfaces you don't want to use.

Be all YubiKey can be

Yubico was good enough to provide us with a YubiKey 5 NFC for this feature. Set up will be different for other hardware tokens, but the result will be the same. We'll use a hardware key as a second factor to log in to our terminals or desktops. This belt-and-braces approach is the same idea as other two-factor authentication (2FA) schemes for web services and apps – text messages being a common second factor. For YubiKeys the required Ubuntu packages are available from a PPA, and other distros will have similar arrangements (check **https://support.yubico.com**, on Arch the required packages are in the main repos). Add the PPA with

```
$ sudo add-apt-repository ppa:yubico/stable
```

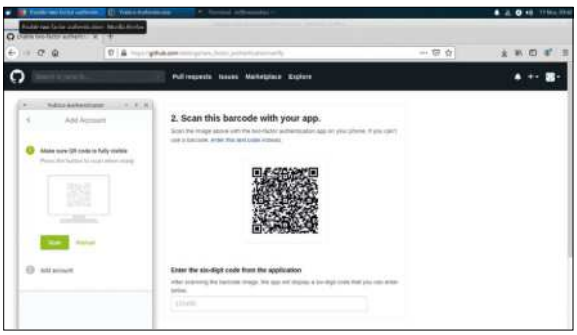
and install the YubiKey Manager and PAM module with:

```
$ sudo apt-get install yubikey-manager-qt libpam-u2f
```

PAM (pluggable authentication modules) takes care of authenticating logins on Linux and allows developers to not worry about how their applications will authenticate with the system. Display managers (such as Gnome's GDM or SDDM on KDE Plasma) as well as the console login all use PAM to authenticate. As the name suggests, modules can be added to authenticate by other means, such as fingerprint readers or, as we've just done, U2F devices.

Plug in the YubiKey. Our first task now will be to associate it with our username, and store the token that generates:

```
$ mkdir ~/.config/Yubico
$ pamu2fcfg > ~/.config/Yubico/u2f_keys
```

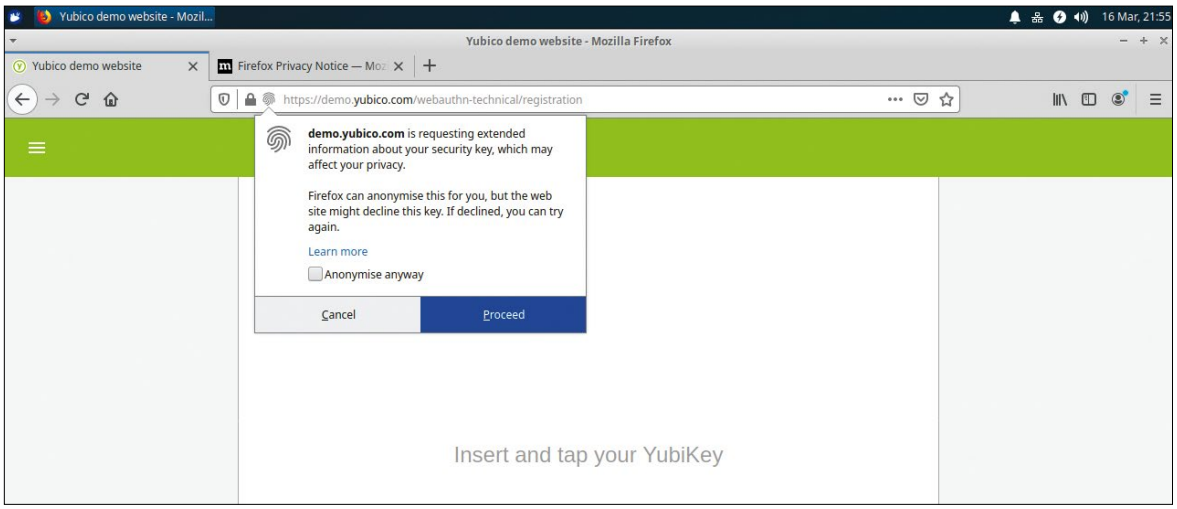


Use your YubiKey to secure your Github logins and keep your code safe. Also keep your recovery codes safe.

» SYSTEMD-HOMED

As we write this the latest *systemd* (version 245) is making its way into the repos of more adventurous distros. Of its many new features, *systemd-homed* stands out, as it (optionally) revolutionises home directories. Traditional static directories work fine for some people and some purposes, but these are problematic for things like network shares and require one type or another of ugly hack to successfully encrypt. *Systemd-homed* also takes care of user management, so that **$HOME**s under its control are completely self-contained and portable. A compliant LUKS2 volume on a USB stick, or with a file of the form **username.home** in the **/home** directory, will be seamlessly mounted in the usual place upon production of the correct credentials. If the system is suspended it will be locked again.

If you want more security in your home directories but aren't yet ready to let go of the old ways, then KDE Vault may be of interest. It enables you to create encrypted vaults that can be stored anywhere, and on production of the correct password are mounted inside your home directory. Their portability means they're ideally suited for use with cloud services such as *Nextcloud*. You don't even need to feel bad for using a proprietary service since they are unlikely to benefit from your scrambled data. The EncFS filesystem doesn't even give away file size information.



Forget U2F. All the kids are doing Authn now. And forget the https padlock, the fingerprint icon is where it's at.

The configuration program is waiting for input, so push the button on the key. If you have other U2F keys, you may want to register those too. With this set up you won't be able to log if your key is lost or burned, after all. For each additional key plug it in, run:

```
$ pamu2fcfg -n >> ~/.config/Yubico/u2f_keys
```

and give it a gentle tap to append that key to our configuration. The **u2f_keys** file can be moved outside of your home directory for additional security, but if you do this you'll need to add the `authfile` parameter in the PAM directives we'll add momentarily. First we'll do a quick safety test, in case the device is malfunctioning, by testing with the `sudo` command. The desired result is that the command will require both the correct password and the YubiKey to be connected. Run:

```
$ sudo nano /etc/pam.d/sudo
```

and then enter your password to look at the relevant PAM directives.

We're going to add a line and then keep the file open (this is what makes it a safety test, so please pay attention to this part) while we test it in another terminal window. This way, if the device or PAM module is misbehaving, we don't need the `sudo` command (which is no longer useful to us since misbehaviour is afoot) to revert these changes. If we didn't discover this until we'd tied our login manager to the YubiKey and logged out, things would be very problematic. Below the `@include common-auth` line add the following:

```
auth      required pam_u2f.so
```

Now save, but do not close(!) the file with Ctrl-O, Enter. Open another terminal and run:

```
$ sudo echo It works
```

You'll be prompted for a password as usual, but if you get it right, nothing will happen until you give your key a tap. If it didn't work, remove the offending line from the still-open file and safely exit, make a cup of tea, and consider available options. If it did work, you probably still want to remove that line. Once we make logging in to our machine with the key mandatory there's less point having it protect `Sudo` as well (unless you enforce a strict post-login key removal and hiding ceremony). Edit the GDM (or whatever display manager you're using on non-Gnome desktops, the syntax is the same) PAM file with:

```
$ sudo nano /etc/pam.d/gdm-password
```

Once again add in the `pam_u2f.so` module below the `@include common-auth` line, and save and close the file. Log out of the desktop and cross your fingers. The display manager should start as normal, ask for your password as normal, and then... do nothing. There's no prompt, but at this point you should tap the device. If you don't do so within 10 seconds, you'll get a not necessarily correct error about incorrect passwords.

There is perhaps some security by obscurity offered by this slightly jarring user experience. Even if the device remains plugged into the machine at all times, an evil maid (*you really should fire them–Ed*) or other visitor to your quarters might, despite somehow knowing your password, not know there was another piece to the puzzle. If you are more disciplined, removing the device whenever you log out and placing it somewhere safe, you stand to win some security points.

The passwordless evolution of U2F, FIDO2 started in 2018, with a new authentication mechanism, WebAuthn, which you should read about at **https://webauthn.guide**. The WebAuthn API provides support for fingerprint or facial-recognition sensors. For now though, your best bet is to use TOTP. Many websites now allow authentication by time-based one-time passwords (TOTP) provided by hardware tokens or mobile apps such as *Google Authenticator*. Once a shared secret is negotiated (us the current time) between parties, the device generates a 6-8 digit code every 30 seconds. This can be verified on the server, and so long as the clocks remain in sync, the user can log in.

THE BENEFITS OF HOTP

"For situations where relying on clocks is unsatisfactory, there is HOTP, which increments counters on each login."

For situations where relying on clocks is unsatisfactory, there is HOTP, which increments counters on each login. The *Yubico Authenticator* works anywhere Google's does, and is available as a portable Appimage from **https://developers.yubico.com/yubioath-desktop**. There's a mobile app too, and it's worth considering the benefits of using a hardware token that isn't your phone (or embedded in it). Phones have a nasty habit of getting lost, broken or moody.

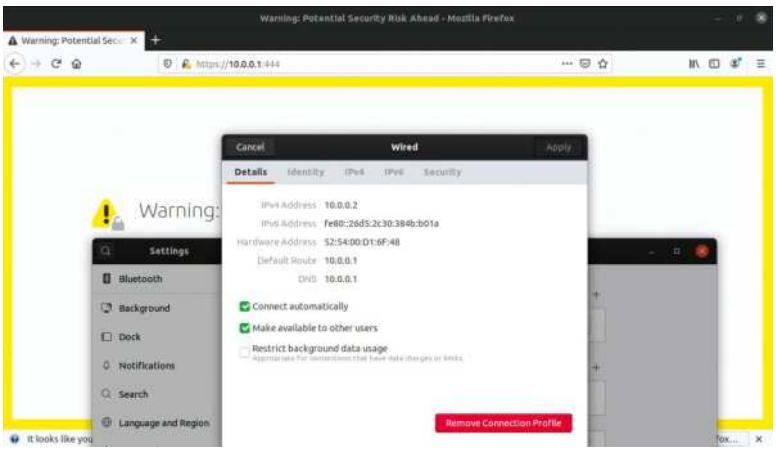
Protect your network

Firewall your home with IPFire, a distro dedicated to keeping your network ports safe.

We covered firewalls last issue, so check that out if you want to learn the ins and outs of packet-filtering on Linux. Now we're going to cover them again, but this time we're going for a more practical approach with IPFire, a dedicated distro for firewalls or other network appliances. You don't need any particularly special hardware to run a firewall, an old machine or a Raspberry Pi is fine (we've included ISOs for 32-bit x86 machines as well as USB/SD Card images for 64-bit PCs and ARMv5 and later devices), but note that at least two network adapters are required. Be that as it may, you can also run IPFire in a virtual machine (which you can add as many virtual Ethernet adapters to as you like). On reasonable hardware and small networks this will perform just fine, although if the host machine can be compromised then so can the virtual firewall, so we lose some security points doing things this way.

If you want to use a VM for IPFire, you can use the 32-bit ISO from the **IPFire/** directory on the disc. If you really want to, grab the 64-bit ISO image from the website. For a small installation it's unlikely to make any difference – memory requirements are low. For a larger or a more complex installation you probably won't want to run a virtual IPFire appliance, although thanks to virtio networking and other bits of virtual voodoo (*see our Virtualisation feature last issue:* **https://linuxformat.com/archives**) this will work fine for simple setups. VPN traffic encryption/decryption requires a fair bit of CPU power, so if you're planning on allowing of data-hungry access to your VPN, be aware of this. You'll find instructions on how to set up a cloud instances of IPFire on AWS and Hetzner cloud on the IPFire website.

Once we set up our Ubuntu VM to use a static IP, we were able to connect to our IPFire VM.



Routing through IPFire

You can route your entire home network through IPFire (by setting it as the default gateway on your home router and shifting settings) or you can avoid upsetting other users of your home network by just routing selected machines through IPFire. Installation (be it real or virtual) is simple, but note that the whole target drive will be erased – the installer provides no means to dual boot from a single drive. Once the system is installed, remove the installation media and reboot to perform initial system setup. Everything is pretty standard – localisation, users and what have you – the important part is the final network setup.

You'll need at least two network adapters, there is no way around this. If you're running a VM you can add a second one with a few clicks and reboot to continue the setup. If you're using a Raspberry Pi 3B or other device with both wired and wireless networking, that will work fine (subject to you setting up an access point with *hostapd*). For a two-adapter setup, we must assign one device the Green network and the other device the Red network. You can use up to four adapters with IPFire, and things get even more colourful if you do that. Use the first option if you need to set up more adapters, and use the second option to assign colours to network hardware. Typically the Green network will be your

IPADDONS

IPFire has everything you need and more to run an advanced firewall solution. But its functionality can be extended far beyond what's in the box. For one thing, it's its own distro with its own package manager (*Pakfire*), which can be used directly or behind the scenes to install extra functionality. We'll talk briefly later about the pros and cons of VPNs, but if you think you need one, you can set it up via OpenVPN with just a few clicks. Two configurations are offered – the appropriately apocalyptic-sounding Roadwarrior, and the more descriptive Net-to-Net. The former may equally have been called client-to-net, and is just what's required for you, a Roadwarrior far outside safe network connectivity, to encrypt your communications back to your trusted server.

Tor, often spoken of in the same sentence as VPNs, can also be set up easily on IPFire. You can set up your instance to access **.onion** nodes and route only your traffic (or only certain parts of it) through *Tor*. Or, if you have the spare bandwidth you can set up a relay and benefit the whole *Tor* community. More conventionally, you can also add a wireless network (usually designated the BLUE interface) to your instance. We mentioned it was possible to do this on a Pi (which has only two network interfaces), but doing it as a third interface saves you having to set up *Hostapd* yourself.

private network and the Red network refers to the one connected to the internet. In practice (if you're not using IPFire on a machine that connects directly to your ISP) these will both connect via your home router, but your Green network interface will connect (via crossover cable, wireless or another router switch) to the machines you want IPFire to protect. The idea is that traffic can flow from Green to Red, but not the other way.

IP addresses must be set up for the network devices under IPFire's control. In the configuration described above, where we have a secure network 'underneath' our home LAN, the Red interface ought to conform to the rest of the LAN, so could be in the form of 192.168.0.something, and the Green interface can technically be anything you want, but it's sensible to use another designated-private address such as 10.0.0.1 or (192.168.1.1 if you prefer). The Red interface (in this setup) can be set to receive an IP address via DHCP, which offers the easiest setup, but you'll probably want to configure a static IP later otherwise you'll be chasing your IPFire instance after a reboot. Static IP will require you to set the gateway to that of your home router. If you're running IPFire virtually then DHCP will use your hypervisor's NAT network, which should work fine.

Unless you want to mandate anyone using your private network use Static IP, the Green interface will need a DHCP server. Turn this on and use the following settings (or something like them):

Start address: 10.0.0.2
End address: 10.0.0.11
Primary DNS: 10.0.0.1

If you're using *libvirt* or *Virtualbox*, this won't work since the virtual NAT device has its own DHCP server, which will get in the way. So you'll have to set up Static IP addresses for the VMs you want IPFire to protect here. For desktop distros, this is most easily achieved by setting a static IP configuration in Network Manager (*see screenshot*). For a physical machine you can connect to the Green interface IPFire host by direct cable connection (older 100mbit cards need a crossover cable, gigabit Ethernet cards do not) or via a switch.

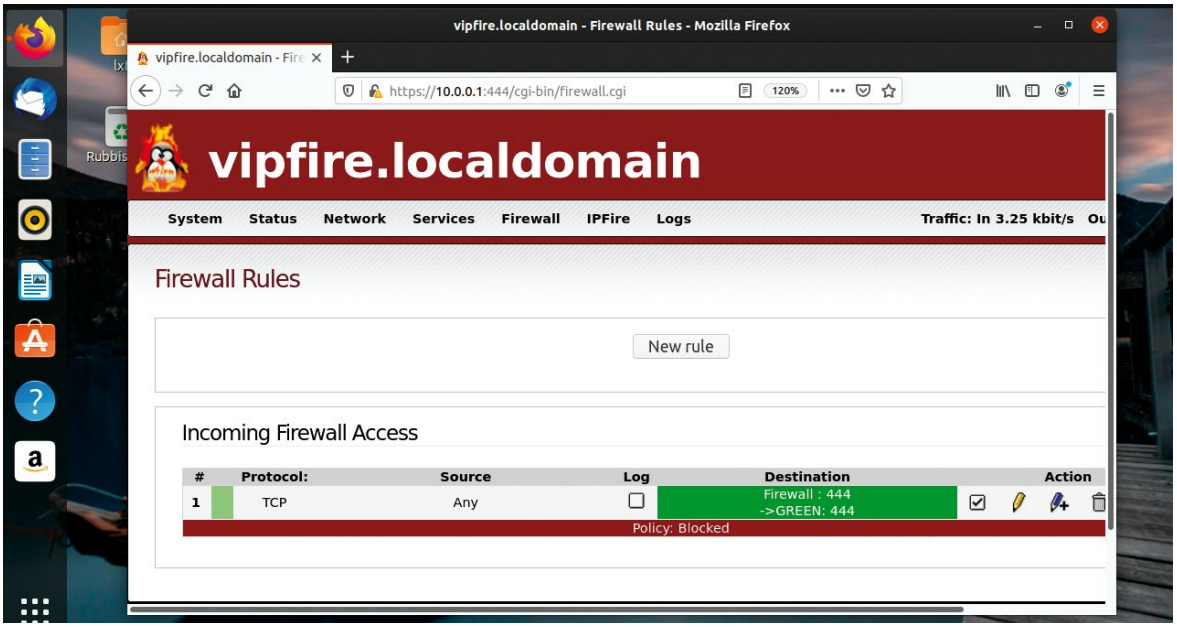
This should be all you need to complete the initial setup of the IPFire instance. You should be able to connect to IPFire from that machine by browsing to **https://10.0.0.1:444**. The first thing you'll see is a nasty security warning, because IPFire uses a self-signed certificate. You can safely ignore this, we promise. The next thing you'll see is a login box, into which you should identify yourself as `admin` using the password you set up earlier. Then you'll be presented with IPFire's intuitive web interface.

By default IPFire forwards DNS requests to the DNS server on the Red interface, which is probably your ISP, via your home router. You may wish to use a public service for this, such as CloudFlare's 1.1.1.1 or Google's 8.8.8.8. This you can do by heading to Network > Domain Name System. Uncheck the Use ISP-assigned DNS box, and click the Add button at the top. Only an IP address is required.

We'll set up a simple rule to allow the Red network to access the web interface on the host. This is not something you'd want to do in real life, but it serves to show the procedure for adding rules. Go to Firewall > Firewall Rules and click the New Rule button. In the Source section, select the Standard Networks option and choose RED. Check the Use NAT box below and choose Destination NAT. In the Destination section select the Firewall option and choose GREEN – 10.0.0.1. In the Protocol section choose TCPm, select Any in the Standard Networks dropdown, and in the Source section enter 444 in the External Port box. In the Additional Settings box, you can choose to log, limit, or rate-limit these connections, but we won't trouble ourselves with that, so just click Add. Click Update, then you should be able to connect to IPFire's web interface from anywhere on your LAN.

IPFIRE DEFAULT FORWARDING

"IPFire forwards DNS requests to the DNS server on the Red interface, which is probably your ISP, via your home router."



Firewall rules look simple with IPFire, and the IPFire interface looks lovely in the Ubuntu 20.04 daily images.

Secure your servers

Rooted or otherwise compromised servers are all over the internet. Don't let yours become one.

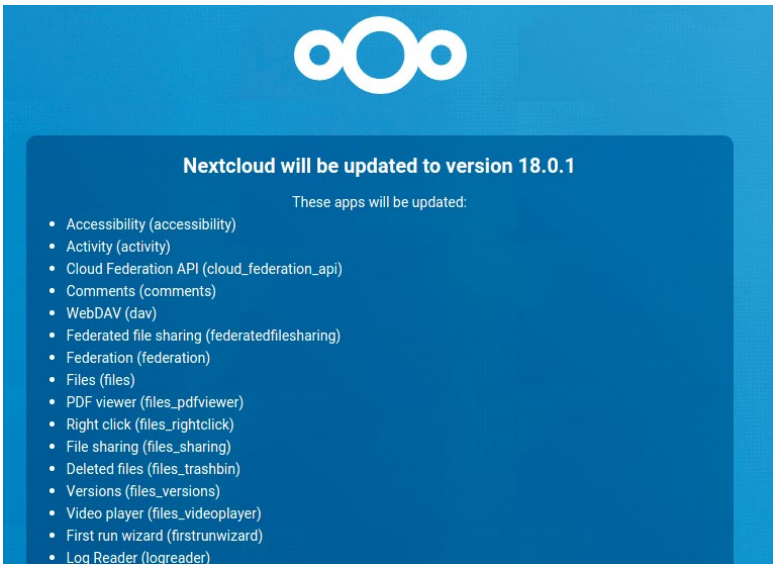
No matter what you do with your Linux servers you will almost certainly have SSH access to them. Indeed this might be the only access you have, so it would be wise to secure it. Naturally, you will already be using a strong password and will have already turned off SSH access for the root account (if you use a login for it). The latter is very important, but generally not necessary on Debian/Ubuntu servers which use `sudo` for elevating privileges. Correcting it on other distros is just a matter of adding

```
PermitRootLogin No
```

to the **/etc/ssh/sshd_config** file and restarting the service. But we can do more. Since we looked at alternatives to passwords on desktops earlier, we may as well study the same topic for servers. Besides passwords, SSH enables you to log in via public key. So by generating a key pair on the machine you log in from, and copying the public part of that pair to the server, you are no longer required to use your password. You may already have a keypair generated, look for a **~/.ssh** directory (on your local machine, not your server) and then look in there for files named **id.rsa** (the private key) and **id.rsa.pub** (the public key). You probably don't want to overwrite these if they're already there, but if you created these a while ago, password-protected them (optional but wise) and forgot the password (these things happen), then you might.

KEYS ADD EXTRA SECURITY

"While the password you set here may be brute-forced or known to an attacker, they still need access to the keys."



Nextcloud. Hopefully it will not go neglected for another three years.

Generating a key pair is easy, just run:

```
$ ssh-keygen
```

on the machine you're going to log in from. You can optionally save the file in a different location, but then you'll need to provide this (via `ssh -i`) every time you use it. You can also provide a password, which might seem counter-intuitive since we are talking about avoiding passwords here. The rationale is that while the password you set here may be brute-forced or known to an attacker, they still need access to the keys for this to be useful. Conversely, if they have the keys, they still need the password, so there's an additional security factor at a cost of only a minor inconvenience to you.

You can add the keys to the server manually, and cloud providers will let you do it from their management interfaces, but it's easy enough to use the following command (again, all this is done on the 'client' side), change servername to where you're logging in to:

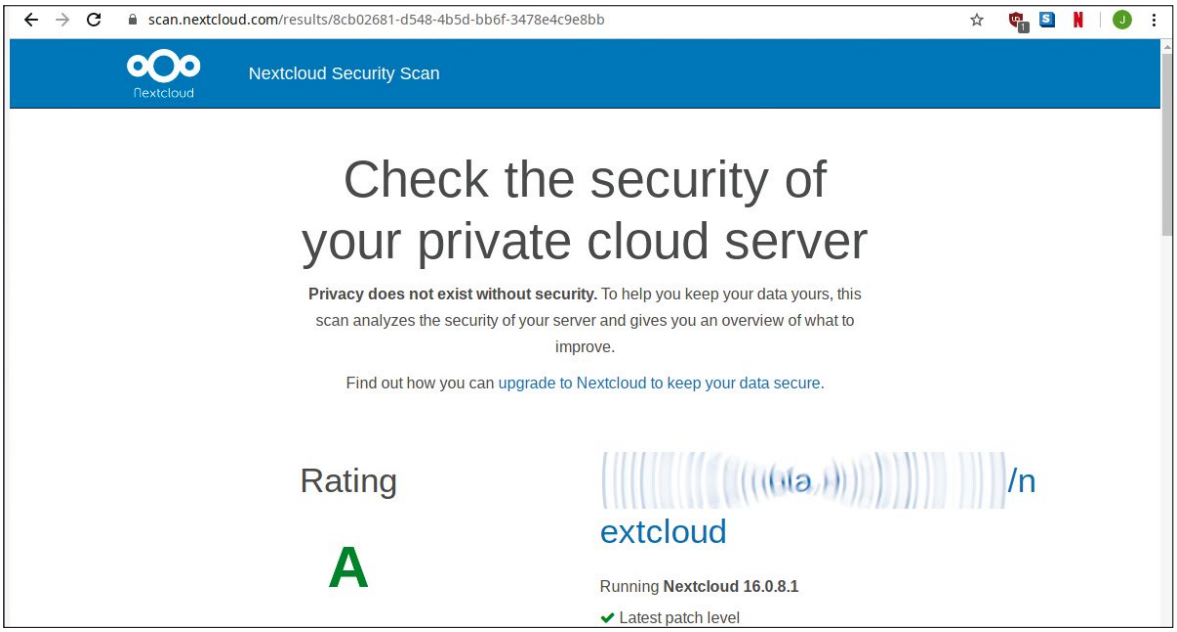
```
$ ssh-copy-id servername
```

This will prompt you for your user password and the key password, before copying the details to the **~/.ssh/authorized_keys** file on the server. It doesn't matter if you have a different username there, once it's in place your next login will only prompt you for the key password.

There's generally no need to try and manage multiple SSH IDs. Keys for multiple servers can be stored happily in the same file. It's also not necessary to try and copy IDs from one machine to another (it won't work). Simply set up a different key pair on each machine you plan on logging in from. Besides using keys, it's wise – and common bread-and-butter security – to set up the *Fail2Ban* service to prevent bruteforce attacks on your SSH logins (for users not enjoying the recently extolled benefits of keys) and other services. We won't cover setting it up here (there's a fine tutorial at **http://bit.ly/lxf262fail2ban**) but the idea is that IP addresses that repeatedly attempt to log in will be blocked (using *IPTables*) for a set amount of time. Not only does this increase security, it also stops your logs being swamped with failed login attempts.

Nextcloud server audits

One of our favourite tools that you can host yourself is *Nextcloud*. We use it all the time at **LXF** Towers because the *Dropbox* client on Linux is awful. However, with great self-hosting comes great responsibility, so it's important to make sure your *Nextcloud* instance is kept up to date and secure. It's easy to forget about these things if they don't break all the time, and indeed this is what happened to us. Sort of. We and other purveyors of fine FOSS have been running *Nextcloud* on Debian for ages – it's a strong combination. Debian will perform unattended updates for critical fixes, and manual



After upgrading, check the security of your instance at scan.nextcloud.com. After this feature's mammoth effort, we get an A!

updates rarely go anything but smoothly. It's easy to update *Nextcloud* from the web interface, it'll even tell you what PHP commands to run to perform the more fragile parts of the upgrade. Running that a few times got us to *Nextcloud 15.0.4*, which is still supported but fairly long in the tooth.

Unfortunately ours was a Debian Stretch install (we did say we've been running it for ages), which only supports PHP 7.0. Since version 16, *Nextcloud* has required PHP 7.1. So there's an important lesson right there: just because your OS hasn't gone EOL yet (Stretch is supported until 2021) software that's running on it may have dependencies it can't provide. In this case we could have used the backported PHP 7.2 packages from **https://deb.sury.org**, but Debian upgrades are usually straightforward and the journey to Debian 10 (Buster) proved painless. So long as you remove all foreign packages from the system beforehand, you should be fine too. The process is well-documented.

Once you have an up-to-date *Nextcloud* there are a few headers the developers recommend setting on your webserver. On Apache, for example, the following should be added to your site's configuration:

```
Header set X-Frame-Options sameorigin
Header set X-Content-Type-Options nosniff
```

The first will stop a modern browser from loading the site in a frame, unless it originates from the same webserver. The second prevents MIME-type sniffing, and prevents external CSS and JavaScript resources from being used if they are not served with the correct `text/html` MIME type. Such requests are blocked, and the option will also prevent HTML, text, XML and JavaScript resources from being loaded externally. For more information on securing your headers, check out the OWASP (Open Web Application Security Project) Foundation's in-depth look at the subject at **www.owasp.org/index.php/OWASP_Secure_Headers_Project#tab=Headers**.

If you have your own VPS, or even your own home server, then you may wish to use it to run your own VPN. These can help you if you're using a connection you don't trust, such as public Wi-Fi. Data is encrypted between you and the VPS server, and then routed onto the destination from your trusted server. IPFire can set this up in a few clicks if this is something you'd like to do at home. You may also subscribe to a commercial VPN service, but despite all the wonderful sponsored articles you'll find offering listicles of the best VPN providers, there's not really a good reason to trust these entities – even the ones with very shiny websites. You're giving them carte blanche access to all your connection data, after all. If your goal is just to hide this data from your ISP, or get around any blocks they have in place, that's fine, but consider why they're blocking that material in the first place, and whether your circumvention of that block constitutes breaking the law.

If you want to read more about server and software weakness in general, we'd encourage you to read *Seven Pernicious Kingdoms: A Taxonomy Of Software Security Errors* by Katrina Tsipenyuk et al. This paper goes some way to categorising threats facing developers and system maintainers today. So stay safe out there. LXF

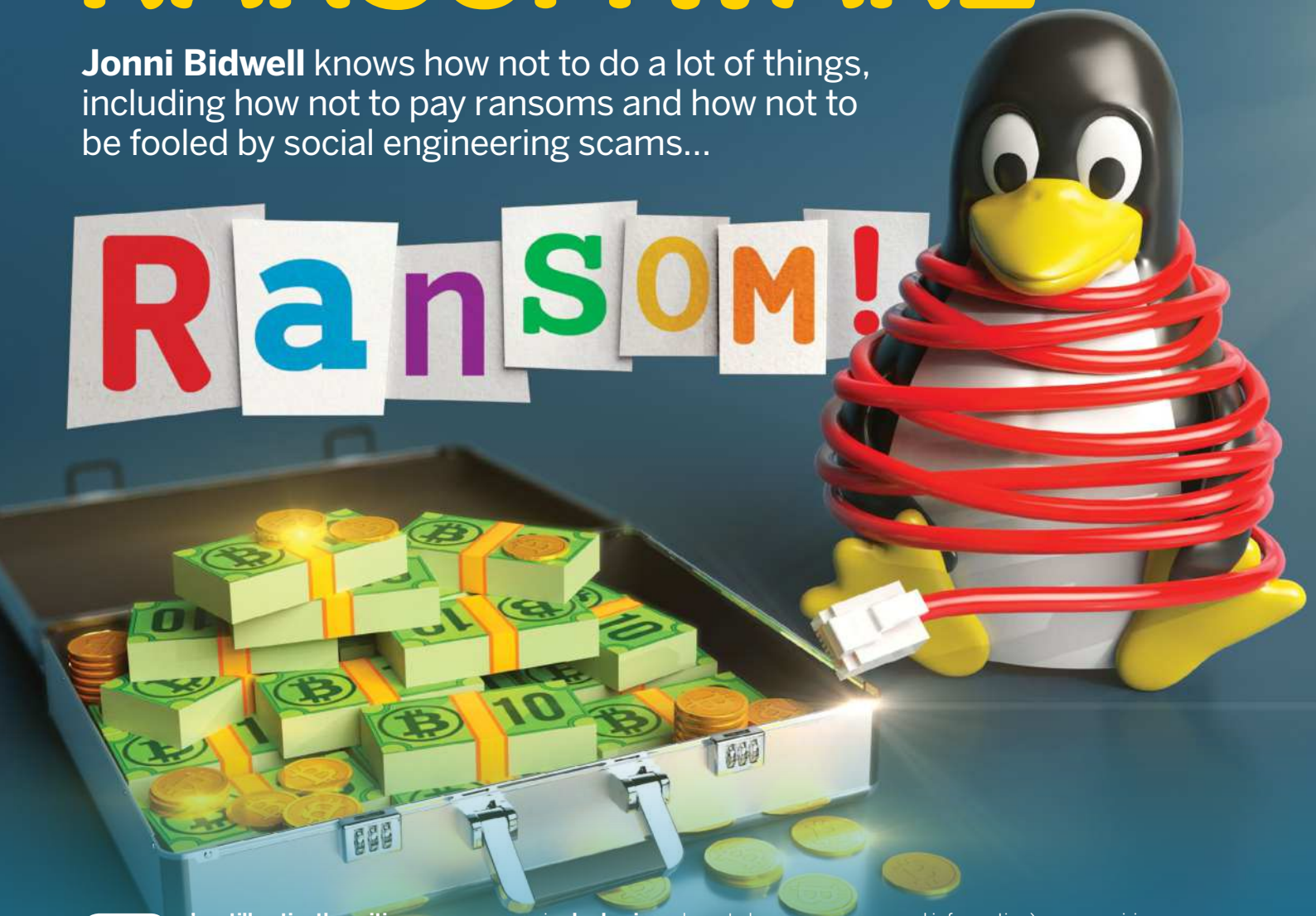
» TO OTP OR NOT TO OTP

If you want to use YubiKeys in your own applications, and issue physical keys directly to trusted users, then check out the developers' guide at **https://developers.yubico.com/OTP**. The *YubiCloud* service provides free authentication for YubiKeys (which come set-up for OTP out of the box), but they also provide open source software so that you can host your own auth server, in the spirit of decentralisation. YubiKeys ship with their own unique AES key, which enables them to be verified by *YubiCloud* (which holds a copy of it). Locally, the symmetric key is locked away in the hardware, and while it's invisible to software it can still be changed.

This is fine if you're doing your own authentication, but even though you can register the new key with *YubiCloud*, the same level of trust is not quite restored, since any new key is subject to being intercepted while it was generated. Because of this, some services won't trust OTP responses from YubiKeys that have had their OTP. You should now be able to test your TOTP setup at **https://demo.yubico.com/otp/verify**. As the site mentions, using a hardware token as single-factor authentication mechanism is not smart, since if it is stolen then everything it has access to could be compromised.

BEWARE OF THE RANSOMWARE

Jonni Bidwell knows how not to do a lot of things, including how not to pay ransoms and how not to be fooled by social engineering scams...



We're still patiently waiting for the year of the Linux desktop, but one sign that Linux has become mainstream is that it is now actively targeted by malware pushers. Perhaps not to the same extent as Windows, and perhaps those miscreants are more interested in hijacking servers than desktops, but peddling the old "Linux doesn't get viruses" line does not make for credible journalism. (*don't use the J word!–Ed*)

Linux servers are compromised all the time. It's a little too easy for your run-of-the-mill script kiddie to find their favourite exploit in Metasploit, find some vulnerable servers via **shodan.io** and create havoc. Attacking desktop Linux is a little more delicate, but that's not to say it doesn't happen. Rogue websites may serve drive-by downloads, bona fide websites may get hacked or a long-standing bug in the kernel may be discovered. All of which might spell the end for your data.

A much greater threat, however, is social engineering. This is pretty much operating system agnostic – it relies on weaknesses in human nature rather than in software – so there'll be a few tips that will help you even if you use some other heathen OS. A well-crafted email can trick a weary mark into opening the wrong kind of file, disclosing their password (or other personal information) or even wiring money to a stranger. Some of these scams are easy to spot, but others are not. Over the last five years, cyber-criminals (we promise not to use this prefix too much) have cottoned on to the fact that people care about their files, so a popular MO is to encrypt them and hold them to ransom. Victims are 'invited' to pay, usually by cryptocurrency, for a decryption key, but often this is never provided. Sometimes ransomware is ill thought-out, and boffins can reverse-engineer whatever encryption was used and provide free decryption tools. But you shouldn't count on this. Instead heed our guide – be strong, be vigilant and behave.

Ransom and deceit

The scammers have figured out that your data is more valuable to you than it is to others. Don't let them exploit this fact.

The best solution to all of these woes is something you should already be doing: regular backups.

While we did a cover feature on malware not all that long ago (**LXF251**), our state of the art analytical engine *(that's me – Ed)* tells us that you want to hear more detail about one particular type of malware – ransomware – and one particular means of delivery, social engineering.

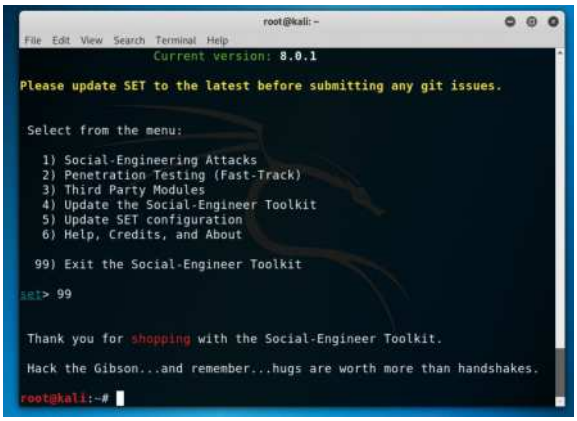
Holding things to ransom is an ancient idea, and one that has been shown to work very well. Kings, nobles and precious works of art have been ransomed since antiquity. Likewise using smooth talking to hoodwink people into doing things they probably shouldn't – grifting, if you will. But in the digital age these practices take on a whole new, sinister dimension. On the face of it having files held to ransom is in an entirely different league to having a family member kidnapped. But what if people's lives depended on the integrity of those files? Terrifyingly enough, this has happened on more than one occasion.

The Wannacry ransomware outbreak of 2017 infected some 300,000 computers in 150 countries – making it, according to veteran security researcher Mikko Hypponen, "the biggest ransomware outbreak in history". People showing up to work were greeted by their computers asking for Bitcoin payments equivalent to $300-$600 to decrypt their files. In the UK it brought our (already struggling–not Wales) NHS to its knees. Patients saw operations cancelled, and were advised to only seek medical care in emergencies. Staff, devoid of network and phone access, had to resort to manual methods to deliver vital healthcare services.

Commentators were quick to point out that the NHS had, and still has, a number of machines running Windows XP, but these weren't really the problem – most of these were embedded installations that can't be upgraded and weren't even connected to the network. The problem was the huge number of unpatched Windows 7/8/2000 systems that were vulnerable to a bug in the SMB protocol of which, it turns out, the NSA was aware for some time. It had in fact weaponised said bug for use in its own Tailored Access Operations unit, and named the exploit EternalBlue.

Unfortunately that exploit fell into the hands of a hacker collective known as the Shadow Brokers. The NSA alerted Microsoft to the possible theft and a patch was issued in March 2017. In April 2017, EternalBlue – alongside other NSA exploits named EternalChampion and EternalRomance – was leaked by the Shadow Brokers and a month later, when many systems remained unpatched, WannaCry used it to wreak havoc around the world. See the box later on for more on the anatomy of WannaCry.

We'd love to say there are magical open source tools that with just a little bit of command-line fu can help you recover from any ransomware attack, but that's just not the case. And nor will it ever be. There are a couple of things that are guaranteed to help you though, and they're skills everyone should practise.



SET, the Social Engineering Toolkit, comes with Kali Linux and is full of tips and tricks (and references to our favourite movie).

» PAY UP OR THE DATA GETS IT

A number of high-profile organisations have paid ransom fees, and we are likely to see a lot more of this unless our collective security game is upped. Large organisations with lax security practices can (but shouldn't) be crippled by a careless employee falling for an email scam. This may be an untargeted attack, where the ransomware keymasters' expected targets are unsuspecting home users, in which case the asking price is pretty modest.

Or it could be some kind of spear phishing campaign, where execs or sysadmins are targeted so their privileges or well-endowed bank accounts can be ransacked. And if you're going to cripple a large organisation you'll probably be after more than pocket change.

In June of this year, Riviera Beach City Council in Florida agreed to pay a $600,000 (65 BTC) ransom to hackers after their systems were crippled. They join a growing number of institutions that have chosen to cough up six-figure sums in order to restore critical services. A 2018 report from SentinalOne surveyed some 500 large organisations and found that of those hit by ransomware, 45 per cent of them had paid up, but only 26 per cent of those had their files unlocked. So much for honour among thieves.

Backup your files

They say there are two kinds of people in this world: people who don't backup their files, and people who have never experienced data loss.

There's no shortage of backup tools available for Linux. *Timeshift* is great for backing up system files. It's vaguely analogous to System Restore in Windows or Time Machine on macOS and it comes as standard with Linux Mint (which is very handy)). For backing up personal files we recommend *DéjàDup* (**https://wiki.gnome.org/Apps/DejaDup**) – it has a simple GUI and can automate everything for you. In its own words "It hides the complexity of doing backups the Right Way (encrypted, off-site, and regular)". It also comes by default with Ubuntu, so check out our handy six-step guide to using it, opposite.

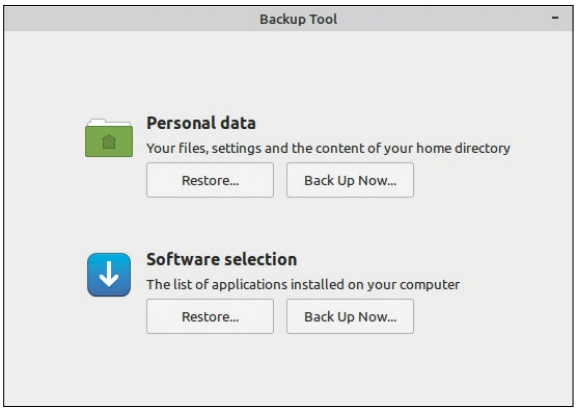
Behind the scenes *DéjàDup* uses the *Duplicity* command line tool, so if you prefer to do things old-school you can hack up some scripts, *cron* jobs and what have you to invoke this directly. All the magic (incremental backups, encryption, scheduling) happens in *Duplicity*, so if your command line and scripting mojo is strong this might be the option for you. It's great for server-type things where you don't necessarily have the luxury of a GUI.

For typical home use your personal files are, we think, much more important than the rest of the system. Whatever flavour of Linux you have, if your system gets hosed you can re-install it and all the applications in a matter of between minutes (Ubuntu) and days (Arch). Fair enough, it might take longer to replicate all the little tweaks that you've added along the way, and maybe it'll never end up quite the same. Your photos and documents and movies are far more precious though.

We like the KeePassXC logo, and we also like when our readers use it and perform regular backups

You may, and should, already back up some of these to some kind of cloud storage, be it an open source offering like Nextcloud (yay!) or a proprietary one such as Dropbox (meh). The trouble with clients like these – ones that automatically keep your local files and cloudy files synchronised – is that in the midst of a ransomware attack they will happily send scrambled files to the cloud. So instead of having an off-site backup you effectively have an off-site rubbish dump.

This all depends on the nature of the ransomware – perhaps it changes filenames, or perhaps the malware interferes with the client and prevents the sync from happening, in which case you would be okay. Still, why take the risk? Use a proper backup tool, and backup to an external hard drive that isn't always connected: cold storage, if you like.



Mint's Backup Tool (not Timeshift) is pretty rudimentary, but ad hoc backups of your important files are better than none

Keeping regular

Backing-up for the first time might give you some sense of security, and rightly so. But backing-up should be a way of life, not a one-off event or even something you do on an ad hoc basis. Unless you integrate it into your digital routine then you're risking all your latest and greatest work (or fresh memes). Back up, back up, back up. Fortunately, thanks to modern tooling this is easier than ever.

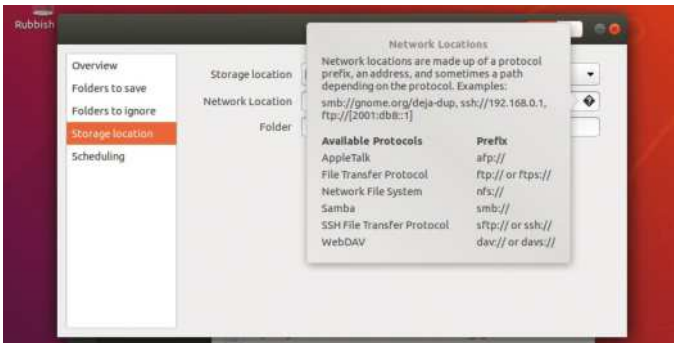
Of course, life will be much easier if your files are in some kind of ordered state, but we can't all be Melvil Dewey *(for one thing, imagine how everyone's address book would look – Ed)*. Your first backup may take a long time, depending on how much stuff you have so pop the kettle on. Or you may have to cancel it when you realise certain directories should have been included or excluded. But once it's done, you're happy with it and don't go messing up your document structure too much, subsequent backups will be much easier. Later backups will probably be much quicker, since they are done incrementally – only files that have changed since the last backup will be stored.

If you're using Mint you may prefer to use its own *Backup Tool* which is simple to use but doesn't have the scheduling facilities. It's great for casual backing-up of your home directory, but doesn't have all the scheduling and other features of *DéjàDup*. You can install it on Mint and other Debian-like systems (you already have it if you're using Ubuntu) with:

```
$ sudo apt install deja-dup
```

Then fire it up and follow this handy guide to securing your precious files.

BACKUP WITH DÉJÀDUP

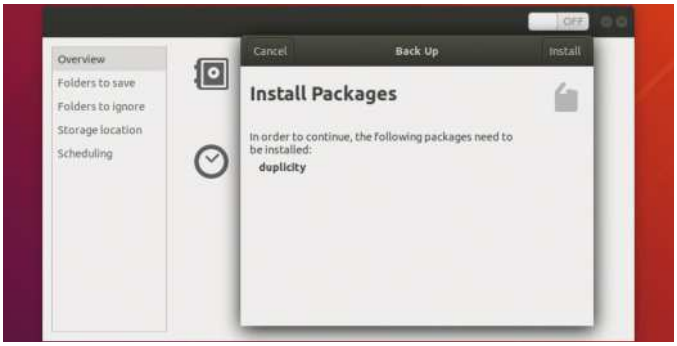


1 Set backup location

You can use Gnome's online accounts feature to back up to Google Drive, a Nextcloud instance and more. You can also use a network location such as Samba or SSH, or a local folder. We recommend using an external hard drive and unplugging it afterwards. If you use a truly local folder, be aware that anywhere you can write to could be hit by a ransomware attack.

2 Choose what to save

Add any folders outside your home directory you want to include in the backup, and exclude any inside that you don't. The **~/.cache/** folder is not necessary and often takes up lots of space – it's where *Chromium*'s cache is stored. If you use *Steam* then you'll want to exclude **~/.steam** as well. Gnome's Disk *Usage Analyzer* is a useful tool to see what's taking up space.

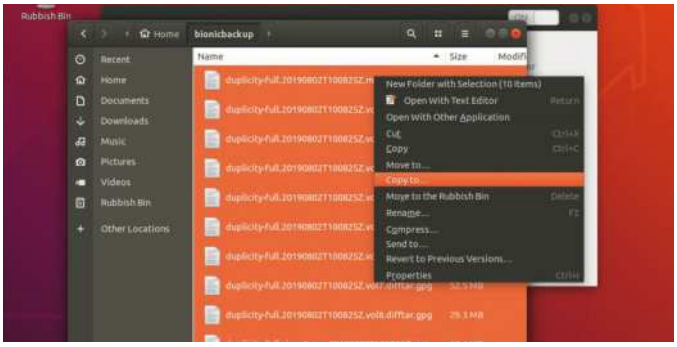


3 Initiate the backup

Hit the Back Up Now button. You may be prompted to install *Duplicity* and you will be prompted for a password. If you use one (encrypting off-site backups is a good idea) then for pity's sake don't forget it; in fact, just use a password manager, like we tell you to later in this feature. Click the Forward button and start to feel secure in the knowledge that your files are safe.

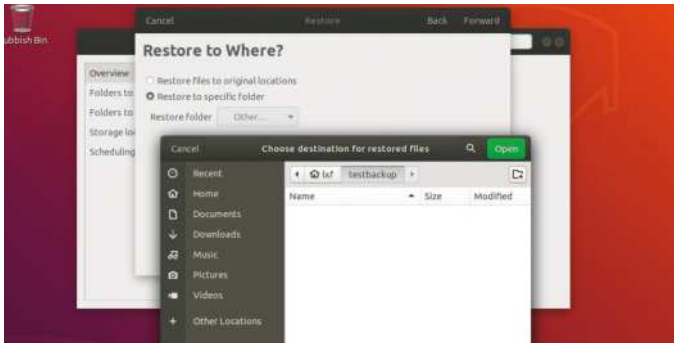
4 Set up a schedule

Just as with yoga, regular practice is the key to backup heaven. Click the Scheduling tab to choose the frequency and longevity of backups. If you backed up to a location that isn't available when an automatic backup initiates, don't worry, you'll be warned. Also beware of running out of space – old backups will be silently removed in this event, which may not be ideal.



5 Be sure to be sure

This step is optional, but a belt-and-braces approach is always good if you truly care about your data. So consider manually copying your encrypted backups somewhere else: a USB key, your home server, somewhere in the cloud, it doesn't matter. Redundancy is good and since you encrypted your backups (didn't you?), losing the storage shouldn't be a concern.



6 Test your backups

Making backups is all very well, but you need to be sure they are viable and that you know the drill for restoring them. Click the Restore button, plug in your storage device if that's how you did it, choose your backup location and choose the date to restore from. The most recent is shown first. For this test, restore to a specific folder so nothing is overwritten.

Anatomy of a ransomware attack

Ever wondered how all this high-tech financial half-inching and data scrambling works? Wonder no more.

Unfortunately, if you fall victim to a ransomware attack and don't have any backups, it's probably too late to do anything about it. The usual asking price to unlock is around $500, though many variants will threaten to increase this amount after a couple of days, and may even threaten to make decryption impossible after longer. These threats just provoke anxiety in the victim, who's then more likely to pay up before considering the merits of doing so.

What is important to remember is that while there's a possibility that whoever's in control of this malware is reasonable enough to provide a decryption key, there's also the possibility that they won't. Their business model, after all, is to collect payments, so why should they bother with the extra admin of sending an email after that is achieved? It's not like people who receive a key will recommend that particular strain of malware to their friends. Be that as it may, whether or not you have access to it a decryption key probably does exist somewhere. There is of course malware that just deletes everything – this class of malware is known as a wiper and would fall into the Chaotic Evil category in D&D – but this doesn't offer criminals much in the way of revenue generation. Since crypto is hard, the people peddling ransomware often get it wrong and it is possible to recover a decryption key without becoming a victim of extortion. To understand how this is possible, it's necessary to understand how all this key generation malarkey *(ha – Ed)* works.

Traditional symmetric encryption relies on two people being able to agree on a secret key by a secure channel. That channel might be a meeting in the woods, or sealed message delivered by a trusted intermediary. Today's numerous, distant electronic transactions do not allow us this luxury. We can't go whisper to our buddy at Paypal every time we want to pay for something online; every transaction requires a new key to be negotiated.

Fortunately we now have public key (asymmetric) encryption which enables us to negotiate a shared secret with a remote other party, be it a person or a machine, even if we've never met them. This is what underlies the key-exchange part of SSL/TLS (used in HTTPS), SSH and PGP communications. Public key crypto is also intrinsic to the workings of all cryptocurrencies. It's a fascinating subject, and also a strange concept to get your head around, but we'll leave you to ponder it at your leisure (or check out our Crypto feature in **LXF247**). All you need to know is that it requires parties to hold not one but two keys – a keypair; that only one of them needs to be kept secret; and that random numbers are involved.

You're unlikely to see a screen like this on Linux, or such a high ransom demand – but you never know.

» MAKES YOU WANNACRY

The WannaCry damage could have been much worse had it not been for the actions of one Marcus Hutchins (aka MalwareTech), a security researcher who noticed WannaCry was trying to contact a server at an unregistered domain. Feeling inquisitive, he duly registered the domain and set up a sinkhole, a server designed to capture information, which had the effect of neutering the malware. Once infected machines were able to contact this domain they stopped trying to infect other machines: in short, Hutchins had found a killswitch.

Further variants of WannaCry appeared in the aftermath, and mercifully killswitch domains were found for these too. Hutchins became something of a hero overnight, which makes the next part of the story quite upsetting. In August 2017 he was in Las Vegas attending the Def Con hacker conference, and was promptly picked up by the FBI on hacking charges relating to the Kronos banking trojan, to which he admitted contributing code as a teenager. In July 2019, Hutchins was effectively granted his freedom, with the judge sentencing him to time already served and even recommending he seek a pardon. This could have gone much worse for Hutchins; the plea deal he accepted could have seen him spend a decade in jail.

We've long commented that technology is moving faster than laws can keep up with. People doing security research have to walk a fine line. They are bound by a treaty known as the Wassenaar Arrangement, by which signatory nations agree to implement regulations governing software that could be used maliciously. The agreement was reworded in December 2017 to make special provisions for security researchers, who may have previously risked prosecution by sharing tools or vulnerabilities across borders.

Ransomware that phones home to some central server could use symmetric encryption to perform its sordid garbling. A random key and a unique identifier could be generated on the target machine and both of these numbers sent and then stored on that command and control (CC) server. All traces of the key would be removed from the victim's machine once files had been duly scrambled, and the unsuspecting victim would then send the requested ransom, together with their unique identifier to the extortionist.

Said extortionist, if they had any shred of decency, would then provide the decryption key and life would go on. This all relies on some degree of centralisation: somewhere out there a list of keys and tokens needs to be stored. For large-scale attacks this isn't ideal – often, multiple command and control servers are used for redundancy – and it makes more sense to somehow store those keys locally. Without public-key encryption this turns into a Catch-22 cascade of futility. You can encrypt the key and store it locally, but then where do you store the key used to encrypt that key?

Instead, an attacker can use a hybrid scheme whereby the symmetric key (or keys) used to scramble all the files is stored on the victim's machine, but encrypted asymmetrically. The simplest implementation of this type of scheme would require our attacker to only look after a single private key, while the public key could be stored locally – inside the decryption program, for example.

So in order to recover their files, the victim needs to pay the ransom, which releases the attacker's private key, which is combined with the public key and these two keys used to decrypt the locally stored symmetric key. Simple, right? Whoever said these people don't work for their money...

Scum, vile scum

Actually, they don't. Ransomware and command and control servers can be readily bought/rented via the darker recesses of the dark web and all this key management business can be taken care of with just a few clicks in a friendly interface. Having a single private key governing all victims is also not a terribly smart idea. It's straightforward enough to protect it as it's sent down the wire from the C2 server to the victim machine over HTTPS or SSH, but a suitably meticulous researcher would, given enough time and a retainer, be able to extract it from memory during the decryption phase. In this way, paying the ransom once (or maybe a few times) could provide decryption capability for everyone. Which is not really what malware authors want and why more advanced key distribution is generally used.

There have been a number of high-profile cases where ransom has been paid and data has been liberated. For example, in 2016 a hospital in Los Angeles coughed up 40 Bitcoins ($17,000 at the time) after being infected with the Locky ransomware. In life or death situations like this, it's easy to see why paying the ransom is a reasonable option. The sum of money involved may well be less than the daily cost of working around a crippled network.

Even if the crooks don't provide a decryption key, the ransom payment may be dwarfed by these costs, so from an accounting (but not awfully principled) point of view the victims aren't that much worse off. Hospitals, and indeed all reputable organisations, ought to have a watertight backup regimen in place, but restoring a whole network under considerable pressure is a complicated business. Consider how long it would take you to restore your machine's software stack, configuration and data after it was wiped, then multiply this number by at least several hundred. Locky evolved over 2016 to become one of the (then) most-detected pieces of malware in the wild, eventually being localised in 30 different languages and no longer requiring C2 infrastructure – which, if it's located in a friendly jurisdiction, is easy enough for authorities to shut down once an outbreak becomes prevalent enough.

Gwarn, put a bullet in our optical drive – we have backups and other PCs y'know.

HOW TO BLACKMAIL PEOPLE

"Ransomware that phones home to some central server could use symmetric encryption to perform its sordid garbling."

Some malware is selective about what it encrypts; it may just encrypt documents (based on file extensions) and leave the rest of the OS intact. Doing this means it only requires user permissions to run. If you're lucky this means it's easy to clean up, at least once the source of the infection is tracked down and so long as backups were taken. Of course, people have come up with ingenious ways of hiding scripts and getting these infections to resurrect themselves. There's nothing more frustrating than thinking you've recovered everything to the pre-outbreak state only to have it all come undone.

'Fileless malware' is becoming the new norm. This often travels by poisoned documents (PDFs in particular) which contain some script or code to fetch hostile code from the internet and inject it directly into memory. This makes it terribly tricky to detect.

Social engineering

Convincing people to do things not in their interests has become quite the artform, so don't be fooled. Click here for puppies...

There are all kinds of devious ways that unscrupulous scammers will use to get hold of unsuspecting marks' data – whether it's tricking them to install ransomware or some other kind of malware such as a keylogger, or having them visit a booby-trapped website and enter valuable credentials. Most people nowadays know better than to reply to an email purporting to come from a Nigerian prince who needs some help organising a wire transfer for his considerable inheritance. Indeed, most scammers no longer bother with these kinds of ruses; their game has been considerably upped.

They may play on local events, especially natural disasters. They may also send targeted email based on public information harvested from the target's social media accounts. For this reason you should be awfully careful about what information you broadcast publicly about yourself. Knowing your interests, location or even those of your friends is enough to gain a data-mining foothold. WannaCry and others like it exploited an unpatched vulnerability in Windows to spread, but this is comparatively rare. Most ransomware, and indeed most unauthorised access to computers, is installed by unwitting users being schemed and duped.

PAY UP OR THE KID GETS IT

"Concerned parents, particularly those that aren't too tech-savvy, will readily part with cash if they think their children are in some sort of danger."

Fraudsters will send email from domains that at a glance look legitimate – say, **linuxf0rmat.com** – or direct you to counterfeit websites at such domains.

KeePassXC's advanced settings are nothing if not impressive, but the defaults are fine. for us

It's incredibly easy to cosmetically clone an entire website with open source spidering tools which have legitimate uses. Widgets and search facilities are harder to reproduce, but if you can dupe victims into thinking the site is real long enough for them to enter their username and password then you've already won. People tend to drop their guard considerably if they feel they are communicating with a friend, so contact lists from plundered email clients are valuable too. Social media accounts then become like gold for scammers.

We'd rather you didn't use Facebook and such, but if you do, it offers a few ways you can secure your account. In particular, you can nominate a person or persons that you trust so that, in the event you get hacked, they will be able to trigger a password reset for you. You'll find these and other options in the Security and Login section of Facebook's settings.

You may have already witnessed hijacked accounts sending messages of the form "Help, I'm trapped in Timbuktu. Please send money", which are easy enough to sniff out. But with a little imagination and a little knowledge about the victim – and if you have access to their social media you'll have plenty at your disposal – a much more convincing con can be crafted. Concerned parents, particularly those that aren't too tech-savvy, will readily part with cash if they think their children are in some sort of danger *(not me – Ed)*.

Human beings aren't really designed to remember passwords or indeed come up with secure ones, so it should come as no surprise that password-reset mechanisms or forgeries thereof are popular means for hijacking accounts. If you search your email address on Troy Hunt's **https://haveibeenpwned.com** website, you may well find it has appeared on one of the many lists of breached credentials that are now forever in the public domain.

This doesn't mean that email address is compromised, just that an account associated with it has been published – possibly something insignificant and maybe even something that has since been secured. For obvious reasons, the actual credentials aren't listed, so you can't check their validity. But doing a thorough password audit is never a bad idea. If a password you used on a forum 12 years ago that's similar to any you use today is in any way traced to a different account, that account is at risk.

It's a painful process, going through all the places where you may have signed up, resetting passwords or cancelling accounts as appropriate. But it might just save your identity, or prevent your contacts getting phished, or something else beneficent. Rather than trying to remember a bunch of secure and different passwords, it's much more sensible to use a password manager. That way you just need to remember a single

strong password, and have the password manager generate (and remember) hard-to-crack, gibberish passwords for all the sites you use. There are plenty of cross-platform solutions that work through browser extensions and mobile apps, but there's a few that actually support desktop Linux too. Our favourite is *KeePassXC* (**https://keepassxc.org**), which is a fork of the largely unmaintained *KeePassX* (which itself was a fork of the Windows-centric *KeePass*). You can install an older (but still good) version straight from the Ubuntu repos with

```
$ sudo apt install keepassxc
```

or use a snap to get a newer version with

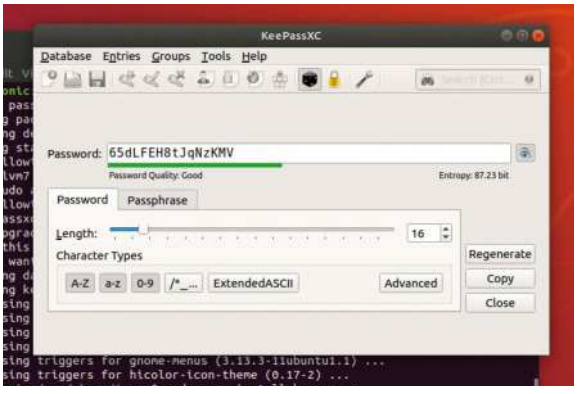
```
$ sudo snap install keepassxc
```

Then fire it up from the Applications menu and click 'Create new database'. The default encryption and storage options are fine for normal humans, but feel free to turn everything up to 11. The next screen is the most important: it's where you must decide your master passphrase. Choose wisely, and consider writing it down somewhere and keeping it somewhere safe, but not somewhere it's likely to get stolen. Choose a location for your password database, and consider backing up this file (and all your important files) regularly. Now add your many, many freshly secured passwords. You can sort them into groups, and install the *KeePassXC* browser extension for *Firefox*.

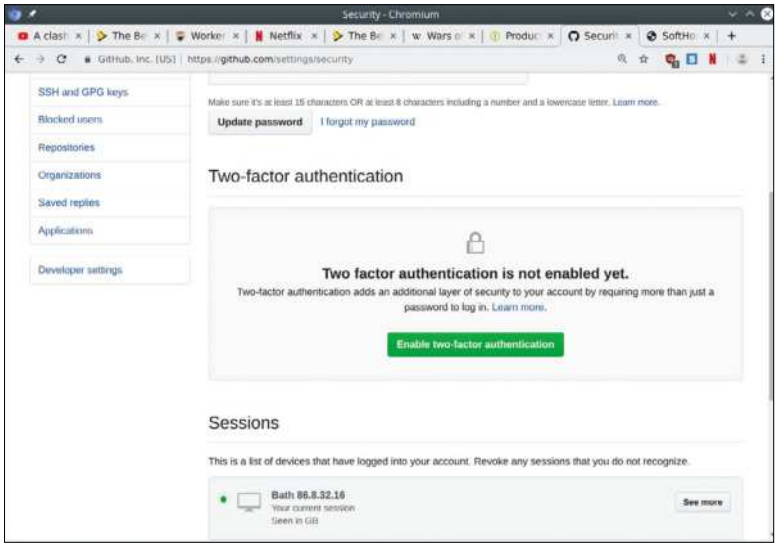
A matter of factor

Many services now offer two-factor authentication (2FA), where besides a password some other token is required. Often this second factor is an SMS message, but there are problems with this. For one, a determined attacker may have the means to carry out a SIM-swap attack, whereby mobile network staff are socially engineered (or bribed or blackmailed) into porting the victim's phone number to an attacker's sim.

Mobile malware is becoming more and more prevalent (don't sideload apps you don't trust!) and if the victim's phone has fallen victim to such then their SMS messages will all be up for grabs. The SS7 protocol that connects cellular networks has long been a source of concern for the security conscious. Once you have access to any cellular network – whether that's internally through a rogue employee or externally by some devious hackery – it's actually possible to access any other network and intercept or reroute messages as befits your whim. So using SMS as a second factor is certainly stronger than using no second factor at all, but other factors should be used for critical services.



Generate unwieldy passwords without worrying about having to remember them.



GitHub supports 2FA. First set it up to use Google Authenticator, then you can use a U2F token.

Hardware tokens and security keys are becoming much more widely supported as a second factor for popular services. The Challenge-Response card readers required by some forms of internet banking are an example; they add credence to the notion that it is the account holder trying to make a given transaction by proving that whoever is doing so has their card and PIN number. Unfortunately, these devices tend to spend most of their time getting lost or running out of battery at the most inopportune moments.

That's the problem with real-world security: it has to take into account all the silly mistakes real world humans are so good at. Stay safe, peeps! LXF

» WE SECOND THIS MOTION

Google's Authenticator app for Android is a software 2FA solution, albeit a proprietary one, that uses the Time-based One-time Password protocol (TOTP) to generate six- to eight-digit one time codes for use as a second factor. It's widely supported, not just on Google sites, so if you have an Android phone this is a free way to boost your security and reduce the chance of unfortunate takeovers. The magic of this is that, once a token is generated and shared between the site and the app, the same code is generated independently based on the current time. The algorithm can tolerate clock skews and network latency, so the app and the site don't need to both keep exactly the same time.

Authenticator codes can theoretically be spied on, though an attacker would have to be watching you in real time for this to work, so for ultimate security we recommend using a hardware solution such as one of Yubico's YubiKey range. These support a variety of authentication protocols. It can work in tandem with a TOTP app as a Universal Second Factor (U2F), or it can authenticate with supported services via NFC (Near Field Communication) on a mobile phone. It can also work as an unlock key for a password manager service. Which is nice.

Yubico kindly sent us some YubiKeys – we'll check them out in a future issue.

VIRTUALISE ALL THE THINGS

Fed up with breaking Linux installs, and Linux installs breaking him, **Jonni Bidwell** looks to virtualisation to ease his very real woes.

Without virtualisation, life at *Linux Format* towers would be a lot more complicated. Testing DVDs would be a nightmare, reviewing new distros would require us to wipe the machine on which we installed previous distros, and if we wanted to test new software on different distros, we'd probably need yet more hardware and yet more time. Yet if you rewind back to the late Mesolithic LXF age – the early 2000s – these were exactly the kind of hardware logistics that the team had to wrangle, all the while living the wild lifestyle encouraged by the heady golden era of dead-tree publishing. Back then tech journalists were made of stronger stuff.

Nowadays things are much more straightforward. If you want to try a new OS, or even if you just want to do something a bit crazy with your current one, all you need do is fire up a virtual machine, and within minutes you have a device that for all intents and purposes behaves like a regular computer. Only you don't need to worry about breaking it – anything you do can be undone, and no one will come at you with pointed questions/sticks if it breaks.

For beginners, a virtual machine is a great way to try Linux. You can run *VirtualBox* for free on Windows or macOS. If you're already running Linux you may prefer to use Red Hat's *Virtual Machine Manager*, which uses *QEMU* (an emulator) and KVM (Linux's powerhouse of a hypervisor) behind the scenes. Whatever your tastes, we've got something for you.

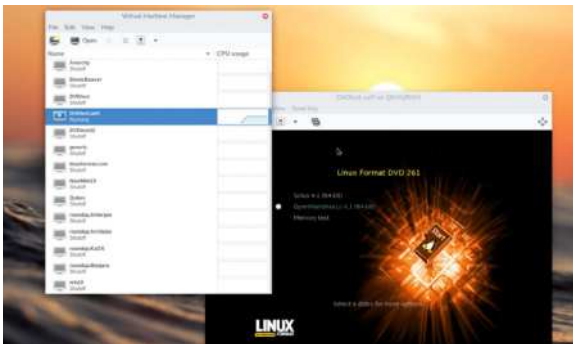
Virtualisation 101

Nobody can tell you what virtualisation is – you have to experience it for yourself. Or you could just read this...

Virtualisation has been around since the 1960s. Of course, computing then was all done on mainframes and OSes were a lot different, so it's harder for youngsters to get their trendy heads around how this worked. The idea then is essentially the same as it is now: compute resources were to be shared (fairly) amongst users in such a way that concurrently running jobs would not interfere with one another. Operating system kernels used to be called 'supervisors', and each separate job was more or less its own entity (today we expect our OSes to multitask programs as a matter of course, but this wasn't the case back then). So the underlying OS which governed these jobs was referred to as a hypervisor, a term still used today.

Modern hypervisors such as Xen and Microsoft's Hyper-V are thin OSes that run on bare metal with the sole purpose of hosting guest VMs strictly and securely, much like the time-sharing schema of the mainframe days. These are called Type-1 hypervisors, which correctly implies the existence of Type-2 hypervisors. The latter, exemplified by *VirtualBox*, *Parallels* and *QEMU* run on a regular operating system and are probably more familiar to everyday users. Things are not binary, though; the Linux's KVM (Kernel-based Virtual Machine) doesn't fit nicely into either category, since it turns the kernel into something like a Type-1 hypervisor, but the host OS still runs as intended.

In 2006, Intel and AMD started shipping processors with, respectively, VT-x and AMD-V extensions. These enabled operating systems to run virtualised without modification, in contrast to previous approaches such as paravirtualisation (which modified the OS to run in a guest environment) or complex software workarounds. Since then, virtually (ahem) all desktop CPUs have shipped with these hardware virtualisation extensions. And they have evolved to enable not only faster virtualisation, but deeper too, with hardware interrupts, memory management units (MMUs) and onboard graphics – via Intel's iGVT-g on Iris Pro graphics – now being virtualisable.



We use only virtual penguins to test our disc now – ever since that ugly uprising six years ago.

It's even possible to blur the boundaries between physical and virtual; actual hardware can be handed off to a virtual machine and used seamlessly. A popular example of this is running a Windows 10 VM with a second (usually high spec) graphics card. This trick, known as PCIe passthrough, enables Linux users to play AAA games at very close to native speeds. PCIe and the general area of Virtual Function I/O (VFIO) require different CPU extensions, called VT-d on Intel and AMD-Vi on AMD. Some of Intel's overclocker-focused chips (the ones ending in K) lack these.

One new project worth keeping an eye on is Looking Glass (**https://looking-glass.hostfission.com**) which aims to streamline passthrough setup for Windows VMs. In particular, the need for a separate monitor and keyboard is obviated.

» VIRTUALISATION, EMULATION AND CONTAINERISATION

It's easy to confuse virtualisation and emulation. Both are a piece of software masquerading as a machine, and both have been around for a long time. In one sense virtualisation is just a special form of emulation. On Linux, we can emulate lots of hardware; for example, a Raspberry Pi with *QEMU*, an Amiga with *UAE*, or a Sega Megadrive with *BlastEm*. We can also emulate software, using *DOSBox* or *Wine* (we know, Wine isn't an emulator – arrest us) for example.

Emulation recreates everything in software, translating or rearranging foreign code into something the host can understand. Virtualisation differs in that the guest code runs directly on the host, which means it's faster (although sometimes other hardware still has to be emulated which makes it slower again), but also means it's only possible to virtualise architectures that are similar. Around a decade after the virtualisation revolution, a new technology emerged: containers. These were popularised by Docker in 2013 and bring many of the benefits of virtualisation (isolation, efficiency), but use the host's kernel, obviating the need to run a whole guest operating system.

Containers can be seen as OS-level virtualisation, as opposed to machine-level. Since a container need contain only the libraries required to run a particular application, they are eminently portable. They also provide a neat solution if that application only works with a particular version of a library. In the same way that Qubes OS runs applications in different VMs, OSes like Container OS (formerly CoreOS Linux) and RancherOS run everything in containers.

VirtualBox beginnings

Learn the basics of virtualisation no matter which OS you're running, or which OS you want to try.

One of the easiest ways to fire up your first virtual machine is with Oracle's *VirtualBox*. This is free (GPL2-licensed) software available for Windows, macOS and Linux. Mac and Windows users should download it from **https://virtualbox.org**, and Linux users should install it with:

```
$ sudo apt install virtualbox
```

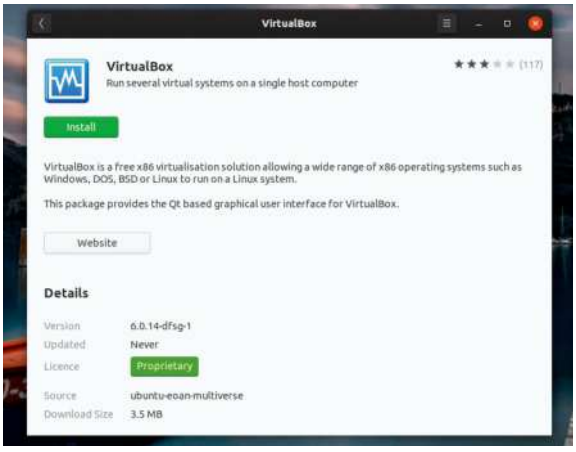
It looks and works the same for all platforms, so no matter what your OS (or which OS you want to virtualise), you can make use of our handy six-step guide opposite. If you're stuck for a distro to try, why not copy the Solus or OpenMandriva ISO files off the **LXFDVD**? If you were to tell *VirtualBox* to use it straight from the disc things would be awfully slow.

If you find yourself stuck in fullscreen mode, use Right Ctrl+F to return to windowed mode. If you find yourself with a mouse cursor trapped in the guest window, just press the Right Ctrl key to escape. Most Linux distributions today support seamless mouse integration, so the latter shouldn't happen to you – taking the mouse past the edge of the screen in the guest should relinquish control to the host.

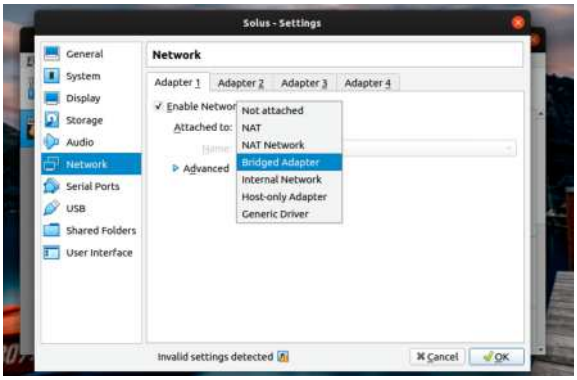
VirtualBox tweaks

It's straightforward to fire up a VM, but let's look at some of the options *VirtualBox* provides to make your virtual life as smooth as your real one. First of all, if you store your virtual drive image on a traditional spinning-rust hard disk, I/O-heavy workloads will be very slow and you'll probably notice a lot of disk activity while they're underway. For such tasks, and indeed if you're brave enough to run a Windows VM, your life will be markedly better if you store the image on an SSD.

If you don't have one, or don't have space on one, you might be able to eke out some extra performance by enabling the host I/O cache for the VM. Select your VM from the column on the left, hit the Settings button on the toolbar, select the disk image (the VDI file, not the optical disc) and tick the box. It's possible to change the type of controller presented to the VM from here, but generally speaking the defaults (PIIX4 for the IDE controller the optical drive is connected to, and AHCI for the SATA controller) are better supported. *VirtualBox 6.1*, released in January, introduced experimental support for virtio storage, which means *VirtualBox* doesn't have to waste CPU cycles emulating a disk controller. This should work well for Linux distributions at least, but will require a driver to be added to a Windows VM. We'll discuss the ins and outs of virtio over the page when we meet *QEMU*.



You'll also find VirtualBox in the Ubuntu Software Centre. It's not really proprietary, don't worry.



Setting up a bridge network (so your guest appears as a separate host on your LAN) is easy, and so are all these other network configs.

The most common thing people want to do is speed up graphics performance. The new default graphics controller in *VirtualBox* (VMSVGA, actually borrowed from VMware) should work well enough for Linux guests. If you have older *VirtualBox* VMs lying around you may want to update them to use the new controller. Virtual Windows may work better with the VBoxSVGA driver. There shouldn't be any reason to use the old VBoxVGA driver these days, unless you're running or trying to virtualise very old distros.

You'll find the Graphics Controller options in the Display section of the VM's settings dialogue. The default Video Memory allocation is only 16MB, which is fine for testing out a Linux distro at sub-1080p resolutions, but for more serious work you'll want to boost this. The aforementioned virtual SVGA devices emulate enough of a graphics card to expose basic OpenGL primitives and some 2D acceleration capabilities – but you'll have probably noticed there's a box for 3D Acceleration too.

If you use this, all the OpenGL capabilities of the host will be available on the guest. The name is a little confusing, because lots of desktop effects which look very two-dimensional rely on 3D extensions. Linux Mint under *VirtualBox*, for example, will tell you it's using software rendering until the 3D box is ticked.

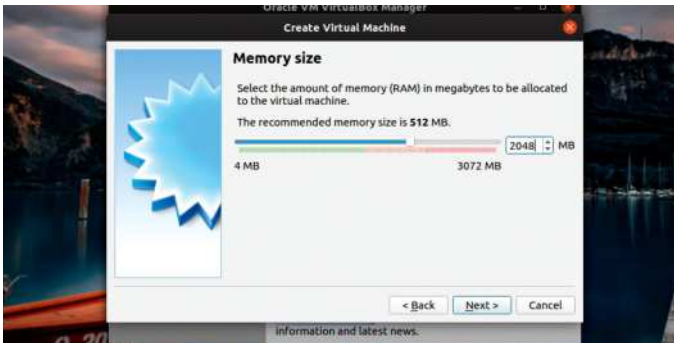
OPEN THE VIRTUALBOX!

1 Get ready

Download *VirtualBox*, as described opposite, and fire it up. By default new VMs are stored in your home directory, so make sure there's some room to play here – or else be aware that you'll need to manually specify a different location. A desktop Linux install that starts at 5GB will very easily grow to 10GB.

2 Create your VM

Hit the New button in *VirtualBox*'s toolbar. We'll use the Solus ISO which we copied from the **LXFDVD** – feel free to use any distro you like, though. Give your VM a name, set the Machine Type to Linux and the Version to whatever is appropriate. The 32-/64-bit selection is important here.



3 Allocate memory

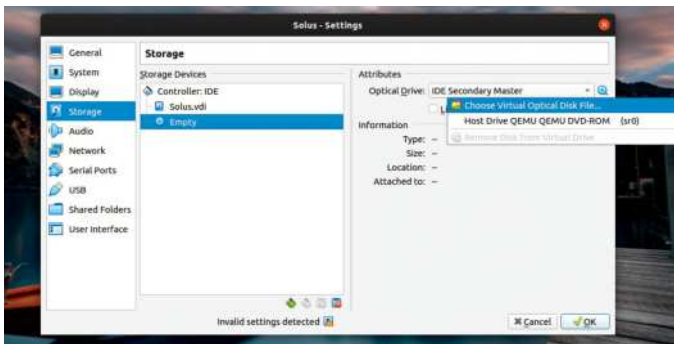
How much memory you allocate to your VM depends on how you want to use it and how much RAM the host machine has. Modern desktops and applications such as web browsers will happily chew through 2GB, whereas booting to the console requires very little. If you allocate too much memory to your VM, host performance will suffer.

4 Sort out storage

We'll need some storage for our VM. Modern Linux distros tend to be fussy about installing to anywhere with less than 10GB of space, and if you can spare it then allocating more is a wise idea. Choose 'Create a virtual hard disk now' (unless you just want to run a Live disc) to begin the process.

5 Tweak storage

Choose the default VDI storage. These files can be fixed or dynamically sized, up to a given maximum. Dynamic storage is obviously much more flexible, and these days you're unlikely to notice the increased I/O overhead, so go with that. You can specify the file's location with the tiny icon to the right.



6 Lift off!

You'll need an ISO image to boot your machine. Go to Settings > Storage and select the optical drive, click the tiny disc drop-down on the right and select Choose Virtual Optical Disk File (sic). Locate your ISO, click OK, then hit the green Start arrow. Chocks away!

QEMU and KVM

Dig deeper and build your own virtual machine at the command line using the power of Linux.

VirtualBox is all well and good, but there are other options. Some people don't like using Oracle software, and in the past the guest extensions (which enabled graphics acceleration and shared clipboards) were not freely licensed, meaning they couldn't be bundled by distros and had to be installed separately.

This is no longer the case and most distros now ship with the required host and guest modules – but occasionally, host and guest modules become mismatched (if a guest distro includes older modules than the host is running, for example) and they must be installed manually. These can be installed – on all platforms – in a running VM by selecting Install Guest Additions from the Devices menu. A virtual CD will be inserted and you'll find a Linux script to build and install the modules, as well as Windows and Mac executables.

Anyway, we digress; we were supposed to be talking about *VirtualBox* alternatives, not providing helpful advice. So, there's the free (but not open source) *VMware Workstation Player*, which is fine for running a single machine on a desktop and from a user's point of view works much the same as *VirtualBox*. However, the virtualisation revolution happened not because people could run clunky desktop operating systems windowed in another clunky desktop operating system, but because powerful servers could house multiple VMs – saving resources, increasing portability and generally making merry. *Workstation Pro* is VMware's commercial offering, and of course there are Hyper-V, Xen and other proprietary tools that have revolutionised infrastructure management. One consequence of this is the rise of cheap VPS (virtual private servers) from the likes of Digital Ocean and OVH. These are a great way to see what remote server administration is all about.

The Linux way of doing virtualisation is a tale of two cities: KVM and *QEMU*. KVM is the hypervisor part and enables the kernel to access the virtualisation functions of the CPU; *QEMU* is the userspace utility that talks to this and emulates virtual hardware. KVM has been ported to many architectures and is today running on Arm, IBM s/390 mainframes and MIPS. *QEMU* is an emulator, and we've already talked about emulation and virtualisation being different animals – but we'll see later that the amount of actual emulating *QEMU* has to do can be reduced to almost zero.

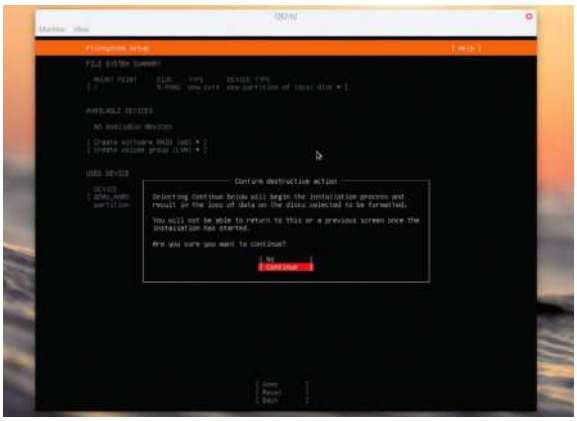
QEMU is a command-line affair, but as Linux aficionados or budding CLI warriors we won't let that get in our way. Installing *QEMU* is just a matter of:

```
$ sudo apt install qemu
```

To deploy our first VM we first need to create a file that can be used to back the virtual storage. *QEMU* supports a number of formats for this; a raw image is the fastest, but the modern qcow2 format is not slow, is more resilient to disk or power failures, and supports snapshotting. So let's use it to create a 10GB image:

```
$ qemu-img create -f qcow2 lxf.qcow2 10G
```

The `qemu-img` command can also convert, resize and do all kinds of other things with disk images, so check the *man* pages for more. As with *VirtualBox*, we need some sort of installation medium to begin. This is an opportunity to grab an ISO file for a distro you'd otherwise be scared to try on bare metal (go on, try



QEMU doesn't look like much at first, but it's incredibly powerful and enables you to perform destructive actions with no risk of reprisal.

» VIRTIO

We mentioned that *QEMU*'s job in the *QEMU*/KVM duality is to emulate hardware, and we've alluded to the fact that this is a relatively slow process. So rather than emulate such things, it makes sense to tell the virtualised OS that it is thus, and that the hardware it can see is not real. In this way we modify the OS (or in this case its device drivers) and have the hypervisor present abstract paravirtualised hardware, which both sides conspire to handle nicely. The framework KVM uses for this sourcery is called virtio, and its drivers enable paravirtualisation of block devices, video cards, network devices and the interestingly named balloon device (used for memory management). The virtio drivers need to be installed on the guest, but have been part of the Linux kernel for a long time – so for Linux guests no action is required:

```
$ qemu-system-x86_64 -drive file=lxf.
qcow2,format=qcow2,if=virtio -vga virtio -net nic,model=virtio -m
1G -enable-kvm
```

If, in the guest, you run

```
$ lspci
```

you'll see your VM now has a virtio-powered video, Ethernet and SCSI disk. In order to take advantage of the improved imaginary video device you'll need to install a desktop environment, which you can do with:

```
$ sudo apt install ubuntu-gnome-desktop
```

Linux From Scratch, how hard can it be?), but we'll be a little beige about it and try out a daily Ubuntu Server 20.04 image. You can grab such a thing from **http://cdimage.ubuntu.com/ubuntu-server/daily-live/current**.

Now we can commence installation, adjusting paths for the install medium and **qcow2** file if they are not in the current directory:

```
$ qemu-system-x86_64 -cdrom focal-live-server-amd64.iso -boot order=d -drive file=lxf.qcow2 -m 1G
```

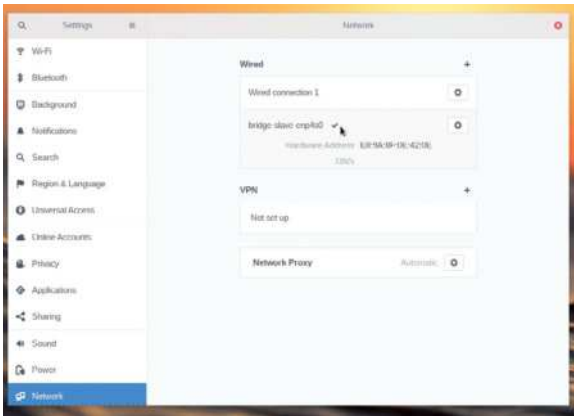
By default *QEMU* assigns only 128MB of memory for its machines, which it turns out is nowhere near enough to boot Ubuntu. The `-m` option at the end assigns 1GB, thus remedying this. A window should open, which you can click in to navigate through the setup questions. By default *QEMU* provides NAT and DHCP services and will set up the VM with an IP address of the form 10.0.2.<number>, which should work fine. We then went with the 'Use Entire Disk' option to avoid needless partitioning. Ubuntu can automatically set up SSH for you, which generally is A Good Thing™, but we'll see later that host-to-guest networking with *QEMU* is tricky. It also offers you a list of popular snaps, so feel free to install anything you think might be interesting. You may want to increase the 1GB of memory and enlarge the image file if you plan to have your VM do anything vaguely onerous.

While the install is underway, or whenever the *QEMU* window is open, you can press Ctrl+Alt+2 to show the *QEMU* monitor, which allows you to assess the state of *QEMU*. Do this and then type `info kvm` to see whether KVM is being used. You'll probably find it isn't, in which case we're emulating a machine, not virtualising one. Oops. That goes some way to explaining the slowness, and it takes only a command line switch to activate KVM. Use the menu or Ctrl+Alt+1 to return to the VM. Once the install has completed, shut down the VM. It does offer to reboot, but go to the Machine menu and hit Power Down instead. As the title bar tells you, use Ctrl+Alt+G to 'escape' the mouse from the VM's window if it's stuck.

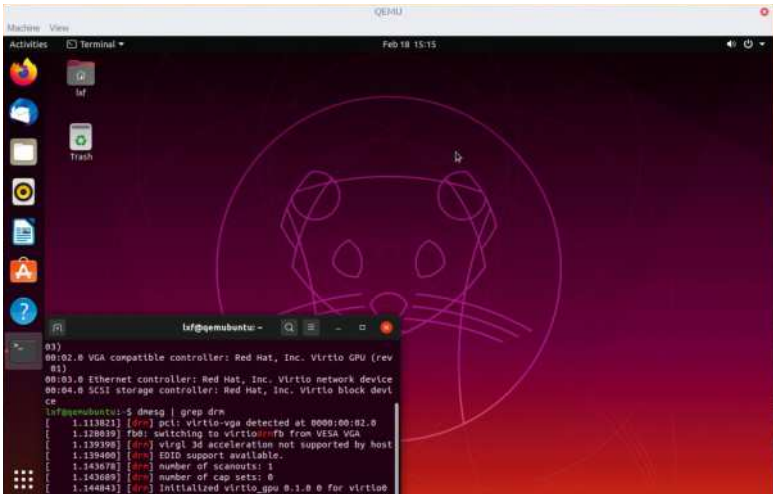
Now start the machine with the correct invocation:

```
$ qemu-system-x86_64 -drive file=lxf.qcow2,format=qcow2 -m 1G -enable-kvm
```

and log in with the credentials you invented earlier. Ubuntu will tell you its IPv4 and IPv6 addresses, but if you try to SSH in from the host, you'll find it doesn't work. *QEMU*'s default network setup enables the guest



If you run into bridging difficulties, it's easy to delete everything and start again from the Network settings area.



Once the virtio drivers were installed, we added 2GB of GNOME goodness to our Ubuntu Server 20.04 VM.

to access services on the host (at the address 10.0.2.2), but not vice versa. Communication between different guests is also not possible in this configuration. We could delve into the ins and outs of setting up a tap device (a virtual network interface) or host-only networking, but since we're going to talk about Libvirt, which makes all these things effortless, we'll defer to the Arch Wiki page on *QEMU*, in particular the lengthy Networking section. See **https://wiki.archlinux.org/index.php/QEMU**.

If you really want nothing more than to have a VM that's easy to SSH into, you can use *QEMU*'s bridge helper to help you (for a wired connection). This involves creating a file called **/etc/qemu/bridge.conf**, consisting of a single line:

```
allow bridge0
```

Now we must create the bridge, add our Ethernet connection (enp3s0 in our case) to it, and connect it to the outside world. If your distro uses *NetworkManager* (it probably does) then it's best to do this with `nmcli` rather than the lower-level `ip` commands:

```
$ sudo nmcli connection add type bridge ifname bridge0 stp no
$ sudo nmcli connection add type bridge-slave ifname enp3s0 master bridge0
$ sudo nmcli connection up bridge-bridge0
```

Then start *QEMU* with:

```
$ qemu-system-x86_64 -drive file=lxf.qcow2,format=qcow2 -m 1G -enable-kvm -net nic  -net bridge,br=bridge0
```

This means your VM will join the same network as the host and get an IP from your home router. The same thing can be achieved in a *VirtualBox* VM by visiting the network settings and changing the adapter type from NAT to Bridged Adapter.

DON'T FEAR THE EMULATOR

"QEMU is an emulator, but we'll see that the amount of actual emulating it has to do can be reduced to almost zero."

Into the libvirt

Putting all the pieces together, we show you how to make your virtualisation faster, easier and safer.

You might gather by now that *QEMU* incantations can get quite unwieldy once you start avoiding defaults – so it's common for people to store these in a script, or use some kind of frontend for *QEMU*. The modern way to wrangle this is with Red Hat's Libvirt. This isn't a frontend – we'll meet *Virtual Machine Manager* soon, which is – but rather an API that provides access to virtualisation functions through its daemon, libvirtd. Libvirt supports a number of hypervisors, including KVM and *VirtualBox*.

This might seem, to the layman, like adding another layer of complication on top of something that's complicated enough. That's a reasonable point; if you just want to set up a VM to try the occasional Linux distro, it's easy to do with *VirtualBox* or any other friendly frontend. Where Libvirt comes into its own is with management. It can take care of storage (resize virtual disks or create pools), networking (no messing around with handcrafting network bridges) and, naturally, can create, start and stop VMs. To some extent it also abstracts the user out of the business of running the VMs, and this is generally good.

Suppose we were running a VM in our favourite desktop environment and something crashed, forcing us to log out. On returning to that VM we'd probably find it in a sorry state, since the process running it (*QEMU* or *VirtualBox*) would have been wiped out.

With Libvirt, VMs – known slightly confusingly as 'domains' – are run by a separate user, so if a local user's desktop crashes while they're playing with a VM, it stays running in the background and can be reconnected to. In this way it works much like connecting to a remote machine. In fact that's another thing Libvirt does: enable VMs to be run from remote machines. The simple *Boxes* tool uses Libvirt behind the scenes to totally abstract away the difference, so through the same interface you can connect to a remote machine, a virtual machine, or indeed a remote, virtual machine.

Installing Libvirt

We can install all the required bits in Ubuntu by installing the *Virtual Machine Manager* (aka *virt-manager*) frontend from the terminal with:

```
$ sudo apt install virt-manager
```

This will also install *QEMU* if you skipped the previous pages (shame on you), and enable the required services. It will also add your user to the libvirt group, but you'll need to log out and back in before you can connect to said services.

Start *Virtual Machine Manager* from the Applications menu, and it should automatically connect to Libvirt; you'll get a permissions error if you didn't obey the previous sentence. Click the 'Create new virtual machine' button at the top left to start the wizard. Select 'Local install media', 'Use ISO image', 'Browse', 'Browse local' (yes, it's a bit long-winded) and choose an ISO file. We went with the desktop version of an Ubuntu 20.04 daily build this time.

Virt-manager will try to guess which OS the medium is going to install, based on the name. This often doesn't work, so clear the 'Automatically detect OS' box, and select Linux and Ubuntu 18.04; it doesn't have to exactly match the guest OS, which is good as not everything is listed. Next, allocate some memory and CPUs to your VM; for a modern desktop distro, anything less than two cores and 2GB of memory is probably going to be painful experience.

Click Forward and choose the 'Create a disk image' option, remembering that virtual machines' disks fill up as quickly as real ones. By default, images are stored in

KEEPING OUR VMS SAFE

The CPU vulnerability can of worms that was opened in January 2018 with Spectre and Meltdown shows no sign of ceasing to spew its vulnerability worms everywhere, with new speculative execution vulnerabilities (CacheOut, Zombieload v2, L1DES and VRS) made public this year. These enable an attacker – albeit by pretty complex means – to access memory that they shouldn't be accessing. One of the major concerns is that on a host running multiple VMs, such attacks could break the isolation, allowing one VM to read the contents of another's memory – even memory protected by Intel's secure enclave (SGX) tech.

Cloud giants host billions of VMs on shared machines, some of which house sensitive data, so this is a grave concern. VMs and the hosts themselves need to have fixes applied and be rebooted in order to be safe, and if you rent a VPS your provider ought to have been in touch with you to tell you this. If it hasn't, you might want to consider changing. In the meantime, the *RIDL Test Suite* available at **https://mdsattacks.com** will tell you which flavours your VM is vulnerable to.

Keeping track of so many CPU vulnerabilities is a full time occupation.

One project that caught our eye as we scrambled to get this feature in on time was Bareflank, a hypervisor SDK. This project is funded by Assured Information Security and aims to enable rapid prototyping of new hypervisors. Bareflank Hypervisor SDK 2.0, including the reference Boxy hypervisor, was released in February. We'd encourage you to investigate further at **https://github.com/bareflank/hypervisor**.

/var/lib/libvirt/images. If you want the image to be created somewhere else – say on another partition where there's more space – then choose the custom storage option here. Go forward and name your new VM – and if you want to use virtual UEFI firmware, don't ignore the tiny 'Customize configuration before install' box. That is the only opportunity you'll get to set this up, using the Firmware option in the Overview section; once the VM has been installed in classic BIOS mode, it is stuck there.

The default NAT Network will work in the same way as *QEMU*'s default set up, but if you want bridged networking, select the vtap device and set the Source mode to Bridge. Graphics performance can be improved by using the virtio video (aka Virgl3d) driver, which enables the guest to use the 3D acceleration of the host. In order to make it work with *virt-manager* you'll need to set the 'Listen' type to 'None' in the Display Spice section. Since we're using the video card almost directly when we enable Virgl3d/virtio, the remote part of this doesn't work.

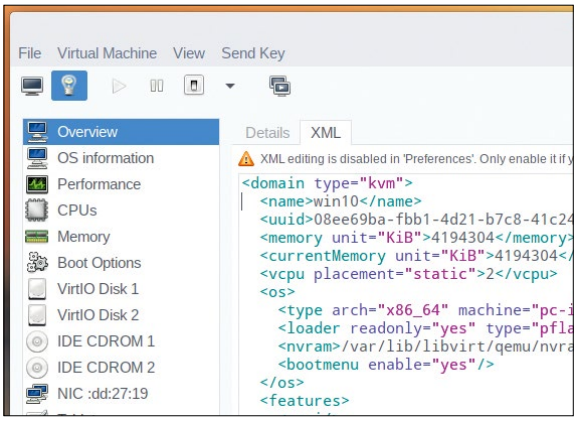
You can check if you've got it working by running `dmesg | grep drm`. You can also easily select virtio network and storage devices. Now you're finally primed to boot your libvirt-powered VM. Hit the Begin Installation button at the top-left, and – fingers crossed – it all works. If not, try the QXL video driver instead (see below).

If you find installing new Linux virtual machines too much effort, then good news – you can download ready-to-go *VirtualBox* and VMware images at **https://osboxes.org**. It's easy to convert the *VirtualBox* VDI images into something *QEMU* can use. Suppose we downloaded the Arch Linux CLI image (and unzipped it to the current directory), then *QEMU* provides the tools to the conversion:

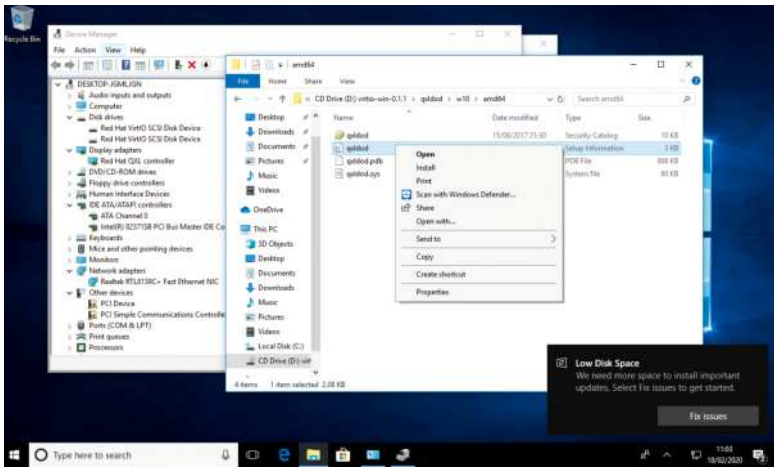
```
$ qemu-img convert -f vdi -O qcow 201905-cli-64bit.vdi archlinux.qcow2
```

Virtual Windows

We haven't covered virtualising Windows here, but it is absolutely possible. You can download an ISO of Windows 10, entirely legally, from Microsoft's site at **http://bit.ly/lxf261windows**. Note that you'll still need to buy a valid Windows key if you want to officially activate the installation once it's all set up to your liking.



Libvirt uses XML to define virtual machinery, and if you need fine-grained control you can edit it directly in virt-manager.



Install the virtio drivers in Windows by 'inserting' the ISO file and right-clicking the relevant INF files.

The singlemost straightforward way to ensure that virtual Windows runs fast is to make sure the virtual drive is stored on an SSD, preferably a fast one. Next, if you're using KVM, you'll want to install the virtio drivers from **http://bit.ly/lxf261virtio**. Scroll down to the direct downloads section and download the Stable (from RHEL) or Latest (from Fedora) ISO file.

It's easiest to do this on the host machine and then use the VM's imaginary optical drive to access it. If you want to use the virtio SCSI driver (highly recommended), you'll need to add this at install time – the rest can be added later. You'll want to give your VM lots of memory; running Windows 10 on less than 4GB is no fun at all. There's no Virgl3D windows driver, but there is an alternative. The QXL video card is a partially paravirtualised video device which should offer better performance (and higher resolutions) than the standard VGA driver. You can use it in *QEMU* with `-vga qxl`, or you'll find it in the Video section of virt-manager. You'll find a Windows driver for it on Red Hat's ISO in the **qxldod/** directory.

> **JUST CAN'T BE BOTHERED?**
> "If you find installing new Linux virtual machines too much effort, then good news – you can download ready-to-go VirtualBox and VMware images."

For the absolute best video performance, though, you'll want to look into PCIe passthrough. With this setup, we install a dedicated GPU just for the Windows VM, and have the kernel isolate it for us at boot. We covered this in our last virtualisation extravaganza back in **LXF244**. One of the greatest resources on this topic is the Arch Wiki page at **http://bit.ly/lxf261passthrough**. If you get this working, we'd love to hear from you and your benchmarks. Virtualising old versions of Windows is possible too, with the correct ISO, which is a reasonable way to hold on to your old apps – perhaps from your now-retired Windows 7 install – until you find suitable Linux alternatives. LXF

Dozens of step-by-step tutorials for beginners and seasoned pros alike

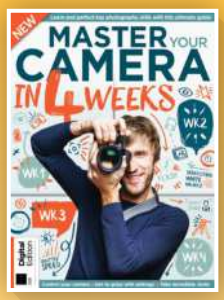
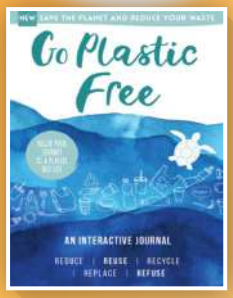
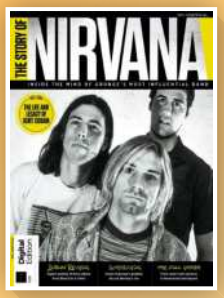
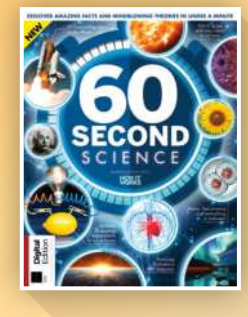
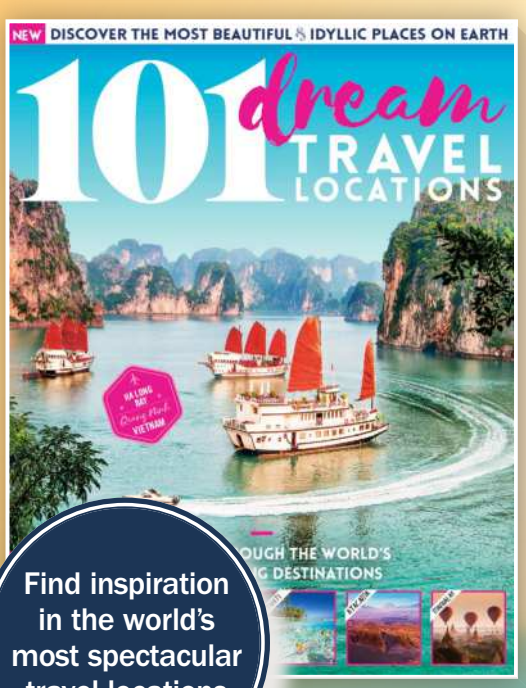


Chart the rise of Hitler and learn how he set Germany on the path to war

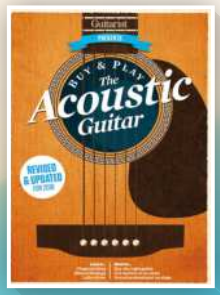








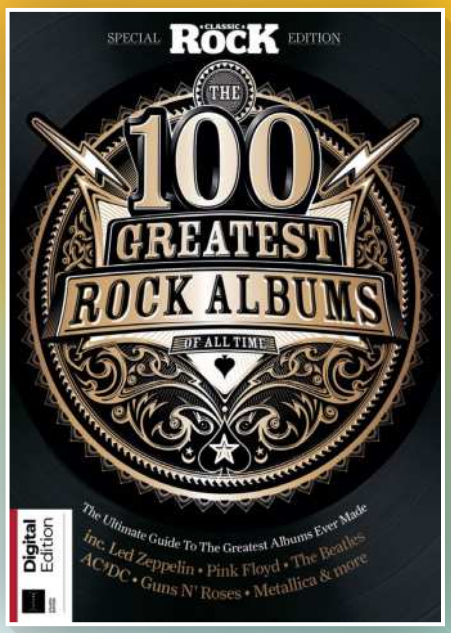
Find inspiration in the world's most spectacular travel locations

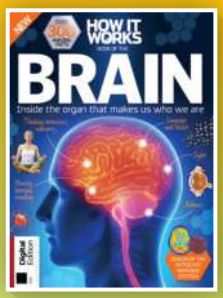


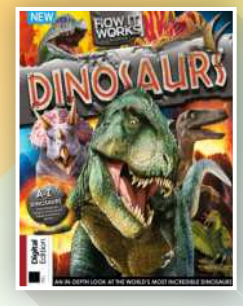
Get great savings when you buy direct from us

1000s of great titles, many not available anywhere else

World-wide delivery and super-safe ordering







DISCOVER OUR GREAT BOOKAZINES

From crochet and quilting to painting and Photoshop, pick up a book that will take your hobby to the next level

Rediscover some of the greatest movies ever made

Follow us on Instagram @futurebookazines

FUTURE

www.magazinesdirect.com

Magazines, back issues & bookazines.

TAMING THE CLOUD

Tiger Computing's **Keith Edmunds** reveals how Kubernetes can be used to build a secure, resilient and scalable Linux infrastructure.

When you want to use Linux to provide services to a business, those services will need to be secure, resilient and scalable. Nice words, but what do we mean by them?

'Secure' means that users can access to the data they require, be that read-only access or write access. At the same time, no data is exposed to any party that's not authorised to see it. Security is deceptive: you can think you have everything protected only to find out later that there are holes. Designing in security from the start of a project is far easier than trying to retrofit it later.

'Resilient' means your services tolerate failures within the infrastructure. A failure might be a server disk controller that can no longer access any disks, rendering the data unreachable. Or the failure might be a network switch that no longer enables two or more systems to communicate. In this context, a "single point of failure" or SPOF is a failure that adversely affects service availability. A resilient infrastructure is one with no SPOFs.

'Scalable' describes the ability of systems to handle spikes of demand gracefully. It also dictates how easily changes may be made to systems. For example, adding a new user, increasing the storage capacity or moving an infrastructure from Amazon Web Services to Google Cloud – or even moving it in-house.

As soon as your infrastructure expands beyond one server, there are lots of options for increasing the security, resilience and scalability. We'll look at how these problems have been solved traditionally, and what new technology is available that changes the face of big application computing.

To understand what's possible today, it's helpful to look at how technology projects have been traditionally implemented. Back in the olden days – that is, more than 10 years ago – businesses would buy or lease hardware to run all the components of their applications. Even relatively simple applications, such as a WordPress website, have multiple components. In the case of WordPress, a MySQL database is needed along with a web server, such as Apache, and a way of handling PHP code. So, they'd build a server, set up Apache, PHP and MySQL, install WordPress and off they'd go.

By and large, that worked. It worked well enough that there are still a huge number of servers configured in exactly that way today. But it wasn't perfect, and two of the bigger problems were resilience and scalability.

Lack of resilience meant that any significant issue on the server would result in a loss of service. Clearly a catastrophic failure would mean no website, but there was also no room to carry out scheduled maintenance without impacting the website. Even installing and activating a routine security update for Apache would necessitate a few seconds' outage for the website.

The resilience problem was largely solved by building 'high availability clusters'. The principle was to have two servers running the website, configured such that the failure of either one didn't result in the website being down. The service being provided was resilient even if the individual servers were not.

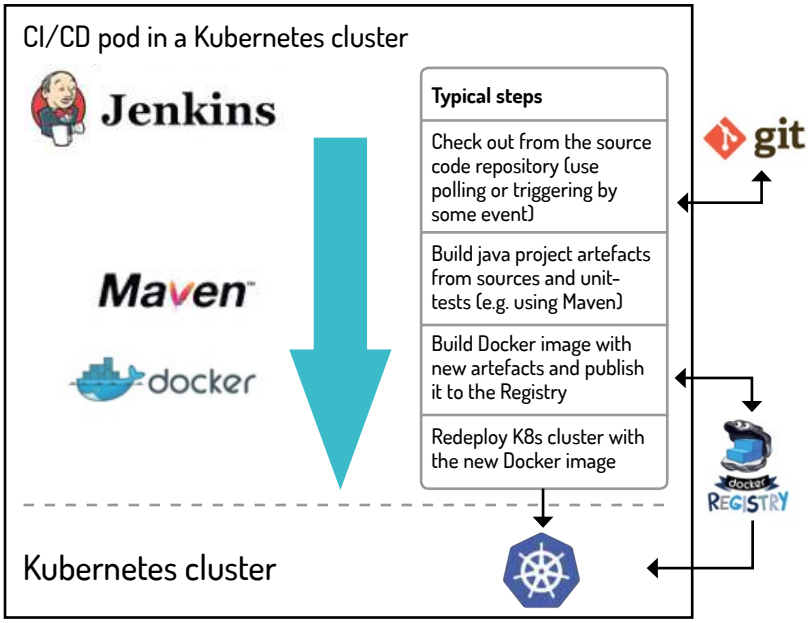
The Slashdot effect

The scalability problem is a bit trickier. Let's say your WordPress site gets 1,000 visitors a month. One day, your business is mentioned on Radio 4 or breakfast TV. Suddenly, you get more than a month's worth of visitors in 20 minutes. We've all heard stories of websites 'crashing', and that's typically why: a lack of scalability.

The two servers that helped with resilience could manage a higher workload than one server alone could, but that's still limited. You'd be paying for two servers 100 per cent of the time and most of the time both were working perfectly. It's likely that one alone could run your site. Then John Humphrys mentions your business on *Today* and you'd need 10 servers to handle the load – but only for a few hours.

The better solution to both the resilience and scalability problem was cloud computing. Set up a server instance or two – the little servers that run your applications – on Amazon Web Services (AWS) or Google Cloud, and if one of the instances failed for some reason, it would automatically be restarted. Set up auto-scaling correctly and when Mr Humphrys causes the workload on your web server instances to rapidly rise, additional server instances are automatically started to share the workload. Later, as interest dies down, those additional instances are stopped, and you only pay for what you use. Perfect... or is it?

Whilst the cloud solution is much more flexible than the traditional standalone server, there are still issues. Updating all the running cloud instances isn't straightforward. Developing for the cloud has challenges too: the laptop your developers are using may be similar to the cloud instance, but it's not the same. If you commit to AWS, migrating to Google Cloud is a complex undertaking. And suppose, for whatever reason, you simply don't want to hand over your computing to Amazon, Google or Microsoft?



Continuous integration and continuous deployment can work well with Kubernetes. Here's an overview of Jenkins being used to build and deploy a Java application.

Containers have emerged as a means to wrap applications with all of their dependencies up into a single package that can be run anywhere. Containers, such as Docker, can run on your developers' laptops in the same way as they run on your cloud instances, but managing a fleet of containers becomes increasingly challenging as the number of containers grows.

The answer is container orchestration. This is a significant shift in focus. Before, we made sure we had enough servers, be they physical or virtual, to ensure we could service the workload. Using the cloud providers' autoscaling helped, but we were still dealing with instances. We had to configure load balancers, firewalls,

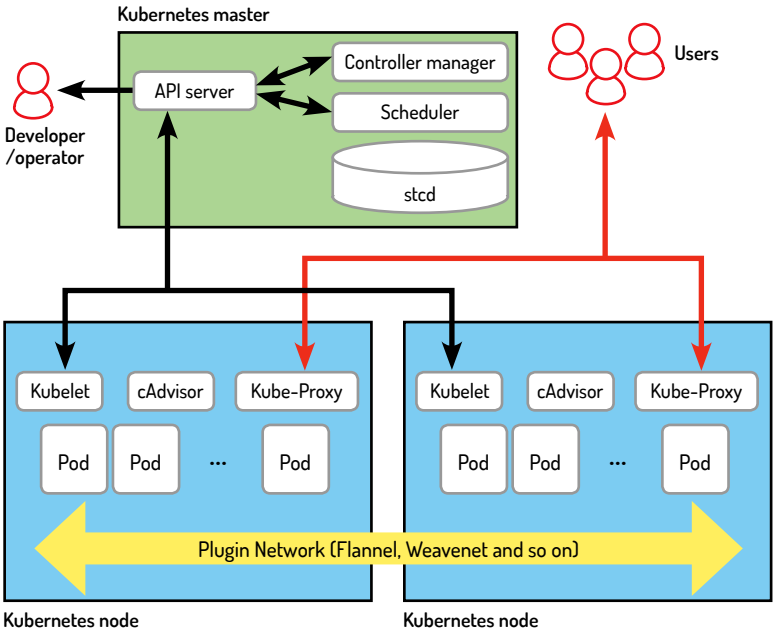
» ABSTRACT CLOUDS

Part of the power of Kubernetes is the abstraction it offers. From a developer's perspective, they develop the application to run in a Docker container. Docker doesn't care whether it's running on Windows, Linux or some other operating system. That same Docker container can be taken from the developer's MacBook and run under Kubernetes without any modification.

The Kubernetes installation itself can be a single machine. Of course, a lot of the benefits of Kubernetes won't be available: there will be no auto-scaling; there's an obvious single point of failure, and so on. As a proof of concept in a test environment, though, it works.

Once you're ready for production, you can run in-house or on a Cloud provider such as AWS or Google Cloud. The Cloud providers have some built-in services that assist in running Kubernetes, but none of are hard requirements. If you want to move between Google, Amazon and your own infrastructure, you set up Kubernetes and move across. None of your applications have to change in any way.

And where is Linux? Kubernetes runs on Linux, but the operating system is invisible to the applications. This is a significant step in the maturity and usability of IT infrastructures.



An overview of Kubernetes showing the master running the key components and two nodes. Note that in practice the master components may be split across multiple systems.

data storage and more manually. With container orchestration, all of that (and much more) is taken care of. We specify the results we require and our container orchestration tools fulfil our requirements. We specify what we want done, rather than how we want it done.

Become a Kubernete

Kubernetes (ku-ber-net-eez) is the leading container orchestration tool today, and it came from Google. If anyone knows how to run huge-scale IT infrastructures, Google does. The origin of Kubernetes is Borg, an internal Google project that's still used to run most of Google's applications including its search engine, Gmail, Google Maps and more. Borg was a secret until Google published a paper about it in 2015, but the paper made it very apparent that Borg was the principal inspiration behind Kubernetes.

Borg is a system that manages computational resources in Google's data centres and keeps Google's applications, both production and otherwise, running despite hardware failure, resource exhaustion or other issues occurring that might otherwise have caused an outage. It does this by carefully monitoring the thousands of nodes that make up a Borg "cell" and the containers running on them, and starting or stopping containers as required in response to problems or fluctuations in load.

Kubernetes itself was born out of Google's GIFEE ('Google's Infrastructure For Everyone Else') initiative, and was designed to be a friendlier version of Borg that could be useful outside Google. It was donated to the Linux Foundation in 2015 through the formation of the Cloud Native Computing Foundation (CNCF).

Kubernetes provides a system whereby you "declare" your containerised applications and services, and it makes sure your applications run according to those declarations. If your programs require external resources, such as storage or load balancers, Kubernetes can provision those automatically. It can scale your applications up or down to keep up with changes in load, and can even scale your whole cluster when required. Your program's components don't even need to know where they're running: Kubernetes provides internal naming services to applications so that they can connect to "wp_mysql" and be automatically connected to the correct resource.'

The end result is a platform that can be used to run your applications on any infrastructure, from a single machine through an on-premise rack of systems to cloud-based fleets of virtual machines running on any major cloud provider, all using the same containers and configuration. Kubernetes is provider-agnostic: run it wherever you want.

ALIEN BEGINNINGS

"The origin of Kubernetes is Borg, an internal Google project that's used to run most of Google's tools"

Kubernetes is a powerful tool, and is necessarily complex. Before we get into an overview, we need to introduce some terms used within Kubernetes. Containers run single applications, as discussed above, and are grouped into pods. A pod is a group of closely linked containers that are deployed together on the same host and share some resources. The containers within a pod work as a team: they'll perform related functions, such as an application container and a logging container with specific settings for the application.

Four key Kubernetes components are the API Server, the Scheduler, the Controller Manager and a distributed configuration database called etcd. The API Server is at the heart of Kubernetes, and acts as the primary endpoint for all management requests. These may be generated by a variety of sources including other Kubernetes components, such as the scheduler, administrators via command-line or web-based dashboards, and containerised applications themselves. It validates requests and updates data stored in etcd.

The Scheduler determines which nodes the various pods will run on, taking into account constraints such as resource requirements, any hardware or software constraints, workload, deadlines and more.

» CHECK OUT THESE RESOURCES

If you're not familiar with Docker (also *see tutorials in* **LXF237**) start here: **https://docs.docker.com/get-started**. There's an interactive, tutorial on deploying and scaling an app here: **https://kubernetes.io/docs/tutorials/kubernetes-basics**. And see **https://kubernetes.io/docs/setup/scratch** for how to build a cluster. You can play with a free Kubernetes cluster at **https://tryk8s.com**. Finally, you can pore over a long, technical paper with an excellent overview of Google's use of Borg and how that influenced the design of Kubernetes here: **https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43438.pdf**.

The Controller Manager monitors the state of the cluster, and will try to start or stop pods as necessarily, via the API Server, to bring the cluster to the desired state. It also manages some internal connections and security features.

Each node runs a Kubelet process, which communicates with the API server and manages containers – generally using Docker – and Kube-Proxy, which handles network proxying and load balancing within the cluster.

The etcd distributed database system derives its name from the **/etc** folder on Linux systems, which is used to hold system configuration information, plus the suffix 'd', often used to denote a daemon process. The goals of etcd are to store key-value data in a distributed, consistent and fault-tolerant way.

The API server keeps all its state data in etcd and can run many instances concurrently. The scheduler and controller manager can only have one active instance but uses a lease system to determine which running instance is the master. All this means that Kubernetes can run as a Highly Available system with no single points of failure.

Putting it all together

So how do we use those components in practice? What follows is an example of setting up a WordPress website using Kubernetes. If you wanted to do this for real, then you'd probably use a predefined recipe called a helm chart. They are available for a number of common applications, but here we'll look at some of the steps necessary to get a WordPress site up and running on Kubernetes.

The first task is to define a password for MySQL:

```
kubectl create secret generic mysql-pass --from-literal=password=YOUR_PASSWORD
```

`kubectl` will talk to the API Server, which will validate the command and then store the password in etcd.

Our services are defined in YAML files, and now we need some persistent storage for the MySQL database.

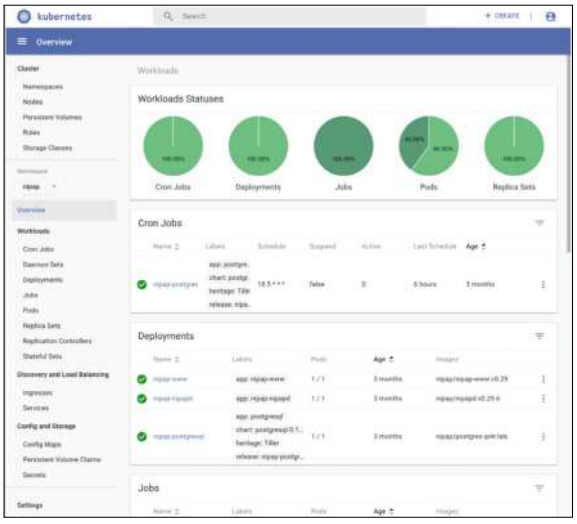
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

The specification should be mostly self-explanatory. The name and labels fields are used to refer to this storage from other parts of Kubernetes, in this case our WordPress container.

Once we've defined the storage, we can define a MySQL instance, pointing it to the predefined storage. That's followed by defining the database itself. We give that database a name and label for easy reference within Kubernetes.

Now we need another container to run WordPress. Part of the container deployment specification is:

```
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  strategy:
    type: Recreate
```



Use the dashboard view to get an at-a-glance summary of Kubernetes in action.

The strategy type "Recreate" means that if any of the code comprising the application changes, then running instances will be deleted and recreated. Other options include being able to cycle new instances in and removing existing instances, one by one, enabling the service to continue running during deployment of an update. Finally, we declare a service for WordPress itself, comprising the PHP code and Apache. Part of the YAML file declaring this is:

```
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
```

Note the last line, defining service type as `LoadBalancer`. That instructs Kubernetes to make the service available outside of Kubernetes. Without that line, this would merely be an internal "Kubernetes only" service.

And that's it. Kubernetes will now use those YAML files as a declaration of what is required, and will set up pods, connections, storage and so on as required to get the cluster into the "desired" state.

This has necessarily been only a high-level overview of Kubernetes, and many details and features of the system have been omitted. We've glossed over autoscaling (both pods and the nodes that make up a cluster), cron jobs (starting containers according to a schedule), Ingress (HTTP load balancing, rewriting and SSL offloading), RBAC (role-based access controls), network policies (firewalling), and much more. Kubernetes is extremely flexible and extremely powerful: for any new IT infrastructure, it must be a serious contender. LXF

PROTECT AGAINST MALWARE

Jonni Bidwell isn't sure if malware could break his PC more than he does, but he doesn't really want to find out...

Linux is no place for malware! "Its small user base means no one's interested in attacking it. The culture of downloading and running random stuff from the web doesn't exist. It's more secure. Ecosystem fragmentation works in its favour, making it hard to target a particular distro", they bellow!

Sadly, none of that is true. Linux servers run most of the internet, there are billions of Android phones, as well as Arm and MIPS-based routers and IoT devices. And people run things of the form `curl... | sudo bash` all the time – it's the recommended install method for a number of popular programs. So the time for Linux complacency is over, and the time for long passwords, diligent opsec and paranoid rants by deluded technical editors is upon us. Don't worry though, we've prepared this comprehensive survey of the threats facing your Linux box and how to mitigate them.

THE RANGE OF MALWARE

"How about software that spies on you via your webcam or microphone?"

Anyone that's used Windows XP knows how annoying adware and nagware are. But what's more concerning is malware that harvests email addresses, passwords, or encrypts your files. If that doesn't scare you, how about software that spies on you via your webcams and microphones? How about malware that steals all your hard-earned cryptocurrency (or real money if you still believe in that)? These threats are all very real, but we'll see that a few simple steps (use a password manager, keep your software up to date, don't open suspect files, be careful which websites you visit) will go a long way towards protecting you.

We'll look at how malware gets its claws in, how it's discovered and what attackers' motivations are. The web is a dangerous place, and more and more we're seeing that web browsers can be compromised irrespective of the platform on which they're running. So we'll do a deep dive on securing your browser. We'll also delve into the (small) world of Linux antivirus.

Malware on Linux

Running desktop Linux certainly puts you in a minority, but malware doesn't discriminate when it comes to your OS.

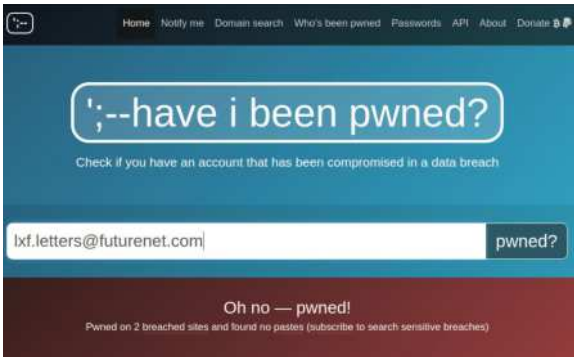
At one stage utterances of the form "Linux doesn't get viruses" were commonplace. They still are, by fans of the operating system and even by distro makers (*see right* **https://manjaro.org**). Unfortunately such talk was never quite true and it's more fallacious now than it ever was. Worms, rootkits and all kinds of other nasties existed on UNIX systems while Linux was still just a twinkle in Linus's eye. Many of the techniques they used were adapted to the Linux platform when it fructified.

As far as desktop Linux is concerned, things are reasonably clear cut: there are far fewer attacks reported, and the list of known malware strains is far shorter than on Windows machines. Given desktop Linux's market share (somewhere around two per cent), this makes sense – attackers will be more inclined to go after whatever the masses are using. In 2006 the 'Get a Mac' series of television commercials claimed "Macs don't get viruses". Since then, its desktop market share has increased to somewhere around ten per cent, and the number of threats has increased proportionally. This in spite of Apple's walled-garden approach to software installation.

These days opinion is divided on whether or not antivirus software is helpful on Windows, not because it doesn't do what it claims to do, but because it sometimes (due to poor programming) opens the system up to new, potentially more dangerous attacks. Security guru Tavis Ormandy, part of Google's Project Zero security team, has been particularly vocal about this threat. Programs that run with the highest level of privilege, as is required for antivirus to scan system files or privileged memory, need to be watertight. If these are vulnerable, then so is the system, possibly more so than if no such antivirus was running.

More often than not you don't need to rely on a vulnerable program to exploit a system: a vulnerable user will do just fine. Writing a program to encrypt all user files, to log/inject keystrokes (at least in X – see **https://github.com/CoolerVoid/rootstealer**), to launch an FTP server (making a machine's files available to the world), even to wipe all local storage if that user has root privileges and can be tricked into entering a password – all this is trivial. The trick is getting a suitably gullible user to run it. Likewise, with some knowledge of a potential mark, a carefully crafted phishing email could trick them into visiting a fake website from whence their passwords can be swallowed, opening a poisoned PDF, or even just running a malicious executable – if somehow that wasn't blocked by either email provider or the target's email client. Humans have a bad habit of trusting things when they (seemingly) come from someone they know, so it's easy to see how these things can cascade and become a large-scale outbreak.

Manjaro's home page says you don't need antivirus software, and they're almost certainly correct.



Oh no, apparently our letters address has ended up on a breach list somewhere. No one tell the readers!

» WE ❤ STATS

The Thycotic Black Hat 2018 Hacker Survey Report[1] states that almost 60 per cent of hackers thought social engineering was the fasted way to pwn a network. Malware (6 per cent) and application vulnerabilities (20 per cent) apparently are much less likely to be used. Thanks to lax security practices and/or hacker ingenuity, there are considerably sized password dumps publicly available. You can check if any of your email addresses are associated with these dumps by visiting Troy Hunt's **https://haveibeenpwned.com**.

A password for an old forum you joined years ago may not be much use to a hacker, but if that forum account can be tied to your identity (via email address, username or otherwise) and you re-use that password or some trivial variation of it elsewhere, the game changes. The survey found that password re-use was by far the most exploited behaviour by hackers (50 per cent), with public Wi-Fi use and poisoned USB drives being joint second at around 20 per cent each.

Of course, the stereotypical image of a hacker doing abstruse things with hex editors, dissemblers and debuggers isn't entirely a work of fiction, but why go to that effort, if just asking nicely for someone's password will do the trick. Of course, these aren't definitive numbers, and the survey reports "300+ participants."

1) https://thycotic.com/resources/black-hat-2018-survey

Protect your browser

Web browsers have become OSes in their own right, and the modern web is a jungle.

Drive-by downloads are pretty rare on Linux, and to be fair they're becoming increasingly rare on Windows too. Gone, also, are the days where respectable programs bundled surreptitious 'optional' extras with their installers, such as browser search bars or spontaneous uncloseable popup adverts. But that's no excuse for complacency.

Where these kinds of things do appear on Linux, it's usually in the form of browser extensions, which are usually installed unwittingly by users. The goodly humans and machine-learn'ed bots that patrol the Chrome store and the Firefox extension page can't catch everything, after all. More often than not, these are easily uninstalled from the Add-ons (*Firefox*) or Extensions (*Chrome/Chromium*) preferences. Recent versions of *Chromium* can log all extension activity; just start it with:

```
$ chromium --enable-extension-activity-logging
```

Then go to the Extensions page, hit Details on the extension you'd like to interrogate, and scroll down to 'View activity log'. If you don't like what you see, disable or delete the extension.

WHEN TO BE PARANOID

"Internet banking is naturally an area in which you should consider using a container, and exercise every caution."

Firefox can refresh your profile without forgetting everything you ever did to your browser.

Firefox's Multi-Account Containers are an extension well worth your attention. As with *Docker* and such, the idea is to segregate workflows, in this case through colour-coded browser tabs. So you can do all your social media, say, in one domain and all your research and actual work in another. They also provide the convenient ability to log into the same site with two different sets of credentials – useful if you manage multiple Gmail or Twitter accounts. Facebook gets its own special container (**http://bit.ly/lxf251fbook**) which ensures that its tracking cookies won't know who you are outside this container.

Firefox containers address privacy concerns more than malware ones (we wouldn't dare slander Facebook by calling it malware), but it's not inconceivable that some advanced malware exists, or will exist, that can use the fact that the user is logged into the high-profile or high-value sites. See more extensions we approve of in the box later on.

Internet banking is naturally an area in which you should consider using a container. And in fact exercise every available precaution: long passwords, regular checks for dodgy transactions, carefully reading any emails purportedly from them for signs of fraud. In particular, check any links to suspect domains. Readers are often wary of our use of **bit.ly** shortened URLs. We think you should trust us (it's impossible to change them after the fact) and oftentimes URLs are longer than a sane reader would dare transcribe, so it's not something we're going to stop. If in doubt, services like **http://checkshorturl.com** or **https://urlex.org** are your friends (but can you trust *them?*)

Freshen up

More persistent browser infections have been recorded in the past, but these days rogue extensions have to fight with a lot of browser safeguards to hide, access files or siphon personal data. That being said, if you are at all worried about your browser then safety trumps sorryness anyday, and since any unwanted browser extras generally live in the user profile directory, an easy fix is just to delete the old profile and start with a new one. Even if you don't think there's anything untoward in there, a fresh profile will have a clean cache and so could speed up your browsing experience.

Firefox even has an option to do almost this straight from the browser. Just navigate to **about:support** and hit the 'Refresh Firefox' button on the top-right of the page. This will preserve your bookmarks, auto-fill information and passwords, but will purge any extensions and themes. Add-ons which live outside the profile folder will survive, but will have their preferences reset. The old profile folder will be backed up to the Desktop and its contents can be copied back to the original location if the situation isn't resolved or you miss something.

If you want to nix your profile directory the old fashioned way, first find out where it is with:

```
$ cd ~/.mozilla/firefox/
$ ls
```

The profile directory will be something like `zx9rh0k.default`. It's a good idea to move this rather than immediately delete it, and it's probably a good idea to backup your bookmarks and make sure you know any saved passwords (or, better, have them saved in a password manager), because this information is all interred in the profile directory. Cookies are also stored in here, so any website preferences will be lost too. Let's move the profile directory to our home directory:

```
$ mv ~/.mozilla/firefox/zx9rh0k.default ~/OldProfile
```

Now restart *Firefox* and see what life with a virgin profile is like. For *Chromium*, the default profile directory is **~/.config/chromium/Default** (or **chrome/Default** if you're using *Google Chrome*). You can check this is indeed where the active profile is stored by visiting **chrome://version**. Substitute this directory in the command above to have a clean *Chromium* profile.

If you're using *Firefox Sync*, or have your *Chromium* profile synced to a Google account, then bookmarks, extensions and passwords are synchronised automatically. This is useful, but in the unlikely event a rogue extension is degrading your browsing experience, using this feature may well re-animate the problem. So do try things without syncing first. If this solves your problem and everything works, then you should

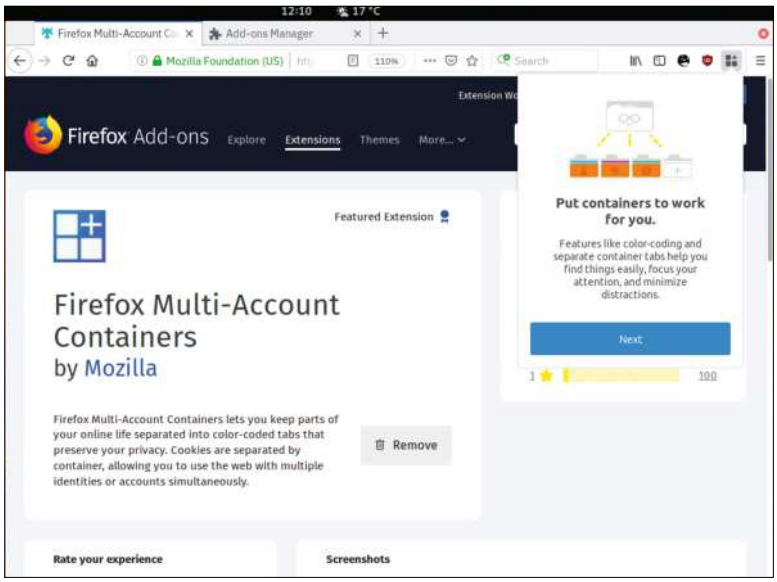
```
$ rm -rf ~/OldProfile
```

because it contains encrypted passwords (and potentially payment information if you use that feature), as well as the keys by which a knowledgeable attacker can decrypt them.

While everyone wishes it wasn't the case, it's always possible that just by visiting a web page you can contract some kind of malware – without so much as clicking, downloading or running anything. Browser developers go to extraordinary lengths to preclude this outcome, and in fact they do a marvellous job because this kind of malware (unless you're running outdated software) is extremely rare.

For it to gain a foothold, there would need to be a vulnerability in the browser, perhaps in a font or image rendering library. Or perhaps a pop-up is able to render in some way that hides a malicious download behind the close button. Generally speaking, an unpatched vulnerability that allowed arbitrary code execution straight from the browser would be quite high-value, so unless you're someone like an activist, political dissident or darkweb mastermind that people with considerable hacking resources (governments, nation states, other darkweb masterminds) are very interested in, then you're probably fine. After all, why would such a group risk losing their 0-day edge on a civvy?

Besides privacy concerns around their tracking cookies (which isn't the focus of this article), advertising networks have in the past been used to spread malware, by injecting malicious JavaScript or Flash applets. There's no reason for the latter to be a problem now: Flash is all but dead and should be uninstalled. Chromium comes with its own PPAPI Flash plug-in which is heavily sandboxed, and by default should prompt before launching. Check this by visiting Settings > Privacy and Security > Site settings > Flash. By all means disable it outright if you want to be sure.

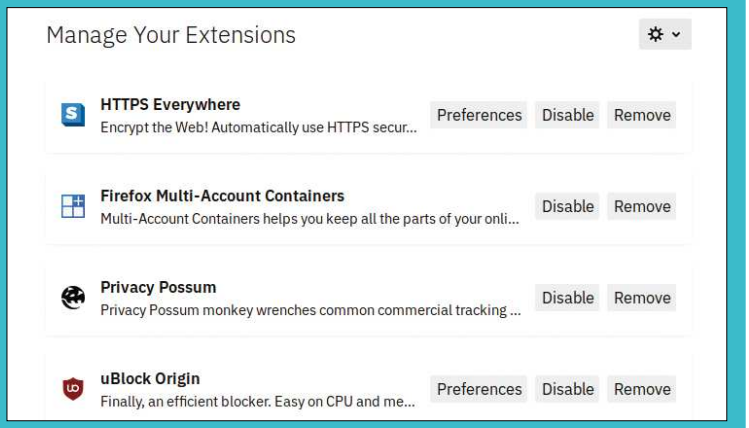


No, you can't run Docker inside Firefox just yet, but account containers can – to quote Wu Tang Clan – protect ya neck.

» BROWSER EXTENSIONS TO USE

All this talk of extensions gone rogue might compel you to get rid of them altogether, but please don't. Some of them are great, and some of them can really help you. For random web browsing, we unreservedly recommend some sort of ad (*don't tell management! – Ed*) blocker. We favour uBlock Origin, but AdBlock Plus (which out of the box will provide a smoother browsing experience without breaking popular sites) is popular too. A close second on the essential extensions list is the EFF's HTTPS Everywhere, which enforces HTTPS transfers, preserving privacy and preventing prying eyes from pilfering passwords. For extreme browsing security, NoScript (for Mozilla browsers) is your friend. This banishes JavaScript everywhere, allowing you to whitelist it piece by piece, or domain by domain. This is bundled with the *Tor Browser*, which is used by exactly the kind of people that repressive regimes target with malware.

Beware though, a lot of sites will behave unpredictably with this NoScript enabled. In particular, it can confuse payment processors, which might leave you staring at a blank page worrying about whether or not you were charged. You probably weren't, but the uncertainty isn't worth it – be sure to disable it before attempting to part with your pennies, such as subscribing to Linux Format. For instance.



Extensions can frustrate your browsing experience, but these ones are awesome.

Vulnerabilities and exploits

Malware begins with breaking programs and ends in tears. Find out how we rein it in.

Without some knowledge of memory management, pointers and execution stacks, it's hard to give a rigorous account of how programs get compromised and thus enable malware to do its thing. This isn't the time or place for such an in-depth look, but we can at least provide an only slightly hand-wavey account of one way a program might be exploited.

Programs work with input, which might come from a user, a website, another program, a piece of hardware (like a keyboard or a sensor) or a file. For a program to work correctly, that input has to be correctly formed. If it's not, then it won't fit nicely into the memory allocated to it, and horrible things can happen. We'll be more specific about what those horrible things are in a moment, so bear with us. So, some kind of input validation ought to take place. If it does not, then the program will attempt to digest something that is the wrong size or shape for the memory assigned to it. As you can imagine, the results of this unpalatability can be unpredictable; to labour a metaphor, you may witness the digital equivalent of anywhere between regurgitation and a food coma. This might be application crashes or system slow-downs.

But it could be much worse. Once such a vulnerability is discovered, it can be exploited by specially crafting our malicious input. *In extremis*, instead of crashing a program by feeding it the wrong kind of cat GIF, we could have it execute the payload of our choosing, which might be a covert keylogger or a persistent backdoor to the host machine. Thus is the progression from vulnerability to exploitation to what is commonly referred to as malware.

For example, if we have a program that asks a user for a number and then spits out the square of that number, we first check that the raw input resembles an integer. If it doesn't, the program quits with an error (we could check for decimal inputs and coerce that to an integer too, but let's keep it simple). Compilers are fussy about data types, and the largest signed integer 64-bit registers can store natively is 2^63-1, so we also need to check if the input is too big (or negative and thus too small) – and if it is, bail with a different error.

There's more than meets the eye to these checks, since the raw input has to exist in some sort of buffer before it can be checked, and how can we know *a priori* how big to make that buffer? Let's not concern ourselves with that – smarter people than us have mostly figured it out. So at this stage we're sure we have an integer, and we want to make our computer square it. That's easy, except for the small (actually large) catch that the result of this squaring this integer could well be larger than the space we allocate for it.

Multiplying an integer with itself will always give a positive integer, so we could use an unsigned int to store our result. That will work fine so long as that result is less than 2^64-1, or, by some elementary mathematical deduction, our original input was less than 2^32. If not, the result may spill into other memory, overwriting it. This so-called integer

ClamAV is developed by Cisco's Talos team.

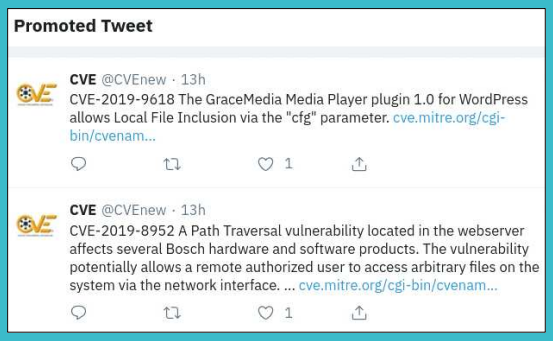
Image credit: "Cisco Systems, Inc"

» CVES, BUGS AND BOUNTIES

There are good ways and bad ways to go about reporting newly discovered vulnerabilities. Up until they're disclosed to the relevant vendor or project, these are known as zero-day bugs, Day Zero being the day that disclosure happens. If you were in possession of a Zero Day, you could brag about it on Twitter (and publish full details on some medium that allows more than 280 characters) – sending affected parties, especially the relevant project, into a panic and possibly unleashing a wave of script-kiddie attacks.

Better is to responsibly disclose the vulnerability privately, so that a fix can be rolled out before things get out of control. Some projects even operate a bug bounty program so that such efforts can be rewarded. Vulnerabilities are assigned an ID on Mitre's Common Vulnerabilities and Exposures (CVE) list, and once a suitable patch is in place, will provide public details of the vulnerability at **https://cve.mitre.org**. This will be picked up and packaged as a matter of urgency by all good distros. Relevant parties may not respond within a reasonable timeframe, or be interested, at which point going public becomes an option.

As we write this, the most recent vulnerability that might affect us is CVE-2019-5953, a buffer overflow in the venerable *wget* – a command line utility for fetching things (most probably webpages) via HTTP. Using the tool to download a maliciously crafted file could result in arbitrary code execution, and a very bad day.



The @CVEnew twitter account is kept busy with so many new vulnerabilities.

The Spectre and Meltdown vulnerabilities were a nightmare for sysadmins, but at least gave rise to these funky, CC0-licensed logos.

overflow is hard to exploit directly, but it has been used historically to attack the SSHv1 protocol (thankfully now deprecated): see **www.kb.cert.org/vuls/id/945216**.

Getting over buffer overflows

Programmers have been trying for 30 years to avoid the kind of oversights that lead to these vulnerabilities and their exploitation. But, to quote our 2015 interview with Mozilla programmer Jim Blandy, "The experiment has been run. Humans cannot be trusted to write that code". Instead of giving up writing code, boffins have come up with new 'memory safe' languages such as Mozilla's Rust, Google's Go and Apple's Swift, which, while not entirely invincible to exploitation, at least put the kibosh on a lot of common memory vulnerabilities.

But rewriting major projects in these languages isn't something that's going to happen overnight, and nor is it an option for large projects (like the Linux kernel) which depend on the features and speed of the C language. It's not all doom and gloom though – these languages are being used productively in a number of places. Mozilla has been developing its Rust-based web engine Servo since 2012, and Go continues to be the language of choice for hipster system programmers. Swift is popular too, but rather than cite a real example we'll direct you to the **@swiftonsecurity** Twitter account where you'll find an irreverent account of information security (infosec) today.

One of the major breakthroughs in defying exploitation became mainstream around 2005 in the form of Address Space Layout Randomisation (the original PaX kernel patch appeared in 2000). This shuffles around the in-memory data required by a program when it is loaded. Before ASLR, an attacker could figure out where a program's resources, such as its stack and libraries, were located in memory. So if the program was subject to some kind of vulnerability, this could be exploited by writing data that would be guaranteed to overwrite, say, the stack.

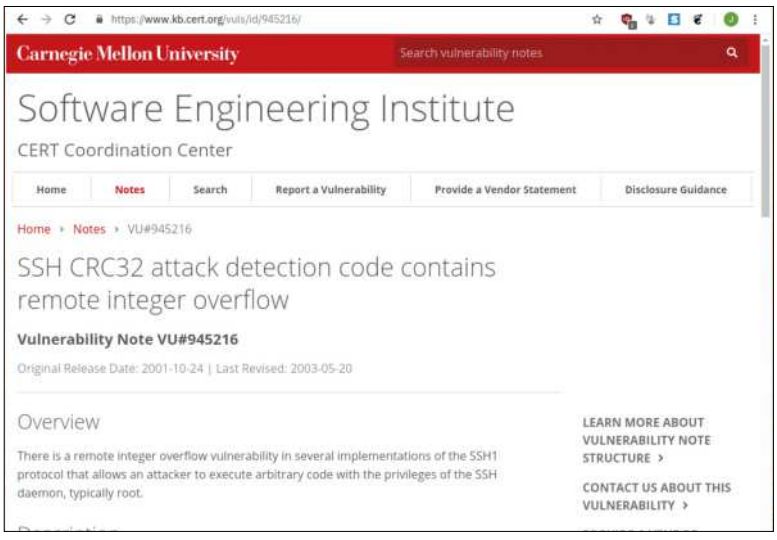
Amid a tide of *Word* macro viruses and email worms this – together with Data Execution Prevention, which prevents designated addresses from being both writable and executable – was great news for Windows. It eliminated a whole class of vulnerabilities. No longer could one just find any old vulnerability and use it to stick any old shellcode onto the execution stack. Unfortunately miscreants moved with the times, and new, more malevolent techniques (such as Return Oriented Programming and Heap Spraying) evolved, and there's no shortage of exploits to this day.

Fortunately tools like *Valgrind* make detecting some of them, at least, slightly easier.

ASLR was extended to the Linux Kernel itself in 2014, making a new acronym (KASLR) and defending against nasty kernel exploits. However KASLR, it turned out, could be defeated using timing exploits on modern CPUs, so further defences were necessary. These came in the form of Kernel Page Table Isolation (KPTI), which was introduced to address the Meltdown vulnerability – though this was kept hush-hush until said vulnerability was made public in January 2018. Meltdown meant that by exploiting a race condition, privileged information could be extracted by processes not entitled to it.

Ultimately it enables a rogue process to read memory from the kernel or other processes, like a password manager. Since it can be exploited through JavaScript, this also means one tab could access information in another, which is pretty terrifying. For more information about Meltdown (and the related Spectre) vulnerability, see our glorious feature in **LXF235** and the latest MDS variant on page 10. New variants of these attacks are being discovered with alarming frequency, but so too are our defences against them. The first round of mitigations came with some fairly serious performance hits to certain workloads, but these are being worked on. Linux 5.1 introduced some new optimisations for the return trampolines ('retpolines') central to protecting privileged memory. With Linux 5.2 a new kernel command line option, `mitigations`, will be introduced.

> **WHY OVERFLOWS WORK**
> "Instead of crashing a program by feeding it the wrong kind of cat GIF, we could have it execute our payload."

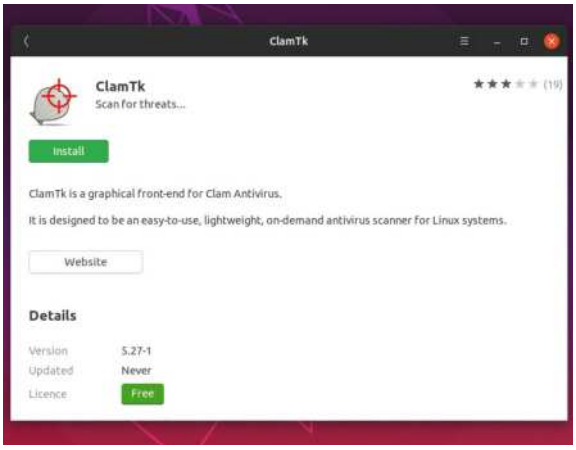


This vintage integer overflow bug actually featured, along with Nmap, in the (disappointing) movie Matrix Reloaded.

The only Linux antivirus...

Your desktop might not need it, but Linux has antivirus software too.

Many of the popular Windows antivirus programs have a Linux equivalent (*Sophos*, *ESET NOD32*, *Comodo*, *F-PROT*). But more often than not these do little more than scan for signatures of Windows viruses. This doesn't mean they should be disregarded outright; if you use *Wine* to run Windows programs then you could inadvertantly use it to run Windows malware. Also, if you run an email server then it's absolutely in your interests to scan incoming messages for Windows threats. Even if you don't, maybe you'd rather know if that file you can't remember downloading contains a Windows nasty, and maybe you'd feel safer scanning it from Linux.



ClamTk can be found in the all-providing Ubuntu Software Centre.

» MALWARE AT LARGE

As the world descends into geopolitical chaos, we hear more and more about the shadowy 'cyber operations' of nation states and the evil genius hackers that reside therein. North Korea, Russia, China and Iran have all been fingered by Western governments for various cyber-wrongdoings. The Wannacry ransomware, which was based around a leaked NSA exploit called EternalBlue, is generally accepted to have come from North Korea.

This nearly crippled health services here in the UK, and could have been much worse had it not been for the deft work of security researcher Marcus Hutchins, who was able to analyse the malware and find a killswitch. Unfortunately Hutchins has recently pleaded guilty to wire fraud charges in the US dating from 2014, when he may have been involved in the shadier side of malware research.

The Russian VPNfilter malware has been detected on half a million routers since its discovery last year. This is has been described by Cisco researchers as a "Swiss Army hacking knife", having a modular structure as well as the ability to persist (via firmware injection) across reboots. One module attempts to downgrade TLS connections so that web traffic can be sniffed and possibly tampered with. A key command and control (CnC) server was seized by the FBI, but infections are ongoing indicating considerable resilience in the botnet. The malware sought out other servers by downloading images from **Photobucket.com**, which had IP addresses hidden in their EXIF data.

We're going to look at what appears to be the only open source antivirus software, *ClamAV*. We actually looked at this way, way back in **LXF131** and to be honest, barring a higher version number, it hasn't changed that much. The Ubuntu 19.04 repositories include a nearly up-to-date version of *ClamAV* (0.100.3 versus the 0.101.2 available on its website). Older versions of the program are available for previous Ubuntu releases, or you can compile it yourself if you're feeling brave – the PPA no longer exists. Generally speaking you're fine with an older version since it will still download up-to-date definitions. Installation is just a matter of:

```
$ sudo apt update
$ sudo apt install clamav
```

The core of *ClamAV* features three main components:

» **clamscan** – a command line tool for scanning files and directories.

» **clamd** – a daemon that runs in the background allowing files to be scanned on access.

» **freshclam** – a tool to update the virus signature database.

The daemon is included in a separate package, so if you're not interested in on-access scanning (which may slow down your system or use lots of memory), don't follow up with:

```
$ sudo apt install clamav-daemon
```

You can run:

```
$ sudo freshclam
```

manually to update the database, but a *systemd* service file is provided to do this automatically. As it turns out, if you try to run that command while the server is running, you'll get an error. Said service can be stopped with the command:

```
$ sudo systemctl stop clamav-freshclam
```

This may be of interest for people that want to set up a *cron* job (in the spirit of **LXF131**). Meanwhile, let's get on with testing our glorious antivirus. We'll download the EICAR test file, which contains a (harmless!) signature that *ClamAV* ought to recognise:

```
$ wget https://www.eicar.org/download/eicar.com.txt
$ clamscan eicar.com.txt
```

You should see output matching the screenshot (*see top right*). For general on-demand use, you can just call `clamscan` with the file(s) or directories you'd like to scan. For delving into directories, or entire filesystems, use the command:

```
$ clamscan --recursive /
```

If you're feeling paranoid, a number of third-party signatures can be downloaded from the repository at **https://github.com/extremeshok/clamav-unofficial-sigs**. These come from various sources (including the Linux-focused Linux Malware Detect, another open source malware scanner), and will increase the chances of false positives, but may also increase your peace of mind.

Windows anti-malware programs are characterised by over-the-top GUIs and paranoid 'Threat Detected' klaxxons. *ClamTk* is a graphical interface to *ClamAV*, but it is much more reserved in its appearance. It comes in its own package, `clamtk`, and can also be found in the Ubuntu *Software Centre*. There are also dedicated Linux programs, a couple of which deserve a mention here, if only because they've been around for a long time: *rkhunter* and *chkrootkit*. These are command-line tools for the detection of rootkits and other nasties.

Secure your servers

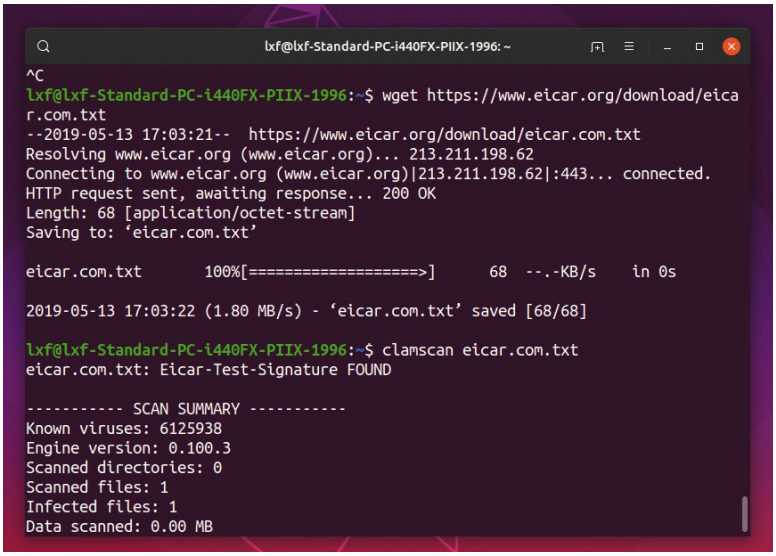
Linux servers offer a much more juicy target than Linux desktops, and potentially are more useful than taking over a Windows desktop machine too. There are a lot of Linux servers on the internet, many of them housing considerable computing power and attached to very fat pipes. So they're ideal targets for cryptojacking attacks or turning into spam-spewing zombies.

Such machines may never be turned off and only rarely rebooted, so if compromised they can be used in long-running attacks. It's quite inconvenient for a hacker if the machine they've compromised suddenly disappears or changes IP address as home machines are wont to do. Sadly, a lot of the Linux servers on the internet live a life of neglect. Deprived of regular updates, they become sitting ducks for script kiddies who have got their grubby paws on the latest weaponised proof of concept code.

System administrators can be socially engineered into giving up passwords or running dodgy programs in the same way as home users can (although they should feel ashamed when they do), but another avenue is to attack the services running on their servers. The open source software that powers popular websites – Apache, Drupal, PHP, MySQL, *WordPress* and pretty much anything else you could care to name – has at some stage suffered from some kind of vulnerability. So has other software found commonly on Linux boxes; vulnerabilities in *OpenSSH* and *Bash* led to more than their fair share of sysadmin hair loss when the Heartbleed and Shellshock vulnerabilities struck in 2015 and 2014.

Besides conventional Linux servers, the burgeoning world of the Internet of Things offers a new kind of (often) Linux-based target. This situation is in some ways opposite to hijacking a server. To an attacker, the server is a single, powerful machine. However, if an attacker finds a vulnerability in, say, a popular brand of home security camera, then they can – through the magic of services like **www.shodan.io**, a search engine for servers and IoT devices – find and take over a sizeable army of these (not very powerful) things. Collectively, they might have a considerable amount of computing power (*hello Bitcoins!–Ed)* or bandwidth which is now at the attacker's disposal.

This is precisely how the Mirai botnet was able to launch a huge DDoS (Distributed Denial of Service) attack against DNS provider Dyn, causing major



A verified European Institute for Computer Anti-Virus Research (EICAR) test signature.

websites to disappear from the internet in 2016. Mirai (*see interview* **LXF244**) was able to become so large because so many devices run an easy-to-find telnet server with well-known, often hardcoded, username and password combinations. The Mirai source code is easy to find online, and new variants of it are still appearing

> **THE THREATS NEVER END**
> "The burgeoning world of the Internet of Things offers a new kind of (often) Linux-based target."

today. In 2017 one was spotted infecting Huawei routers. In 2018, one, known as Satori, even attacked cryptocurrency mining operations through a vulnerability in the popular *Claymore's Miner* software, interfering with payment addresses for nefarious profit.

If security-conscious people had their way, IoT devices would never ship with default credentials, but it seems marketing people have got their way. Coming up with new credentials before you can use your fancy internet-enabled fridgecam (or whatever) is, it would seem, deemed user-unfriendly. It's well worth going out of your way to change any default passwords on webcams, routers, NAS devices and the like.

For devices where these things are hardcoded, the device should be firewalled so that it's not accessible from the outside world – or possibly taken back to the shop. It's also worth running a portscan on your network to see what services are running on what devices. Our favourite tool for this purpose is *Nmap*, which you can find in all good repos, and which will tell you all about your network with something like:

```
$ sudo nmap 192.168.0.*
```

Running it as root thusly allows Nmap to identify devices through MAC descriptors. LXF

Credit: https://github.com/sa2c/sombrero

SOMBRERO

Comprehensively benchmark your PC

Ed Bennett shows how you can apply the same techniques used to benchmark the world's fastest supercomputers to one-up your friends...

OUR EXPERT

Ed Bennett
When not speed-testing supercomputers, Ed helps and trains academic researchers.

QUICK TIP

Running across multiple machines works best if you have the same username on all of them. If you don't, you'll need to put the software in a shared directory, and create a ~/.ssh/config file to let Open MPI know how to connect to each machine.

You've probably seen benchmark statistics reported about all manner of popular devices, but what exactly is a benchmark, and how can you run one on your own systems?

A benchmark originally referred to a testing process for rifles, where the gun was mounted to a bench, rather than being held by a human marksperson: the bench was acting as the mark, hence benchmark. With several test firings, the spread of shots could be measured – the smaller the spread, the better the gun. Nowadays however it refers to any process that allows different methods, devices or processes to be compared with each other directly, based upon a standard reference.

In computing, this is done by taking a set of reference tasks which reliably take exactly the same amount of effort each time, and measuring how a system performs when completing them. These tasks might be designed to stress one specific component – for example, hard disk write speed; the machine, to see what the absolute limits of the system are (synthetic benchmarks); or they might be representative of the kinds of work that a system will be used for (application benchmarks). A gaming benchmark might run a set portion of game over a fixed route and measure the frame rate, while a 3D rendering benchmark may choose a complex 3D scene and measure the time the computer takes to render it. Benchmarks are frequently distributed as benchmark suites; since it's rare for a computer to only be used for a single task, benchmark suites measure all-round performance by including contributions from a variety of typical tasks.

```
[RESULT] Case 1 25.54 Gflops/seconds
[MAIN] Case 2: 224.23e9 floating point operations and 552.88e6 bytes communicated
[MAIN] Case 2: 40.62 operations per byte
[RESULT] Case 2 224.23 Gflops in 9.292359 seconds
[RESULT] Case 2 24.13 Gflops/seconds
[MAIN] Case 3: 303.73e9 floating point operations and 552.88e6 bytes communicated
[MAIN] Case 3: 55.02 operations per byte
[RESULT] Case 3 303.73 Gflops in 14.904015 seconds
[RESULT] Case 3 20.38 Gflops/seconds
[MAIN] Case 4: 515.52e9 floating point operations and 736.10e6 bytes communicated
[MAIN] Case 4: 70.03 operations per byte
[RESULT] Case 4 515.52 Gflops in 22.535351 seconds
[RESULT] Case 4 22.88 Gflops/seconds
[MAIN] Case 5: 1105.36e9 floating point operations and 1104.15e6 bytes communicated
[MAIN] Case 5: 100.11 operations per byte
[RESULT] Case 5 1105.36 Gflops in 44.300056 seconds
[RESULT] Case 5 24.95 Gflops/seconds
[MAIN] Case 6: 2067.93e9 floating point operations and 1840.25e6 bytes communicated
[MAIN] Case 6: 112.37 operations per byte
[RESULT] Case 6 2067.93 Gflops in 75.660873 seconds
[RESULT] Case 6 27.33 Gflops/seconds
[RESULT] SUM 4364.06 Gflops in 172.461 seconds
[RESULT] SUM 25 Gflops/seconds
```

A run of Sombrero on a two-core laptop with an Intel Core i7-7660U takes around three minutes.

In this article, we'll look at the *Sombrero* benchmark suite. *Sombrero* is a suite of application benchmarks for a piece of high-performance research software used in theoretical particle physics computations. The benchmarks within it stress to varying degrees the raw speed of the processor, the speed at which data can get into the processor from the system memory, and the speed at which data can be communicated between the different CPU cores and computers that are cooperatively working on a problem.

Despite some effort to promote alternatives, in the world of high-performance computing the command-line reigns supreme. Because of this, the instructions that follow all take place at a terminal. To start, you'll need to install the dependencies: *Git*, *make*, *GCC*, *bc*, *gnuplot*, *OpenSSH* and *Open MPI*. On Ubuntu, these can be installed with:

```
$ sudo apt install git make gcc bc gnuplot \
  openssh-server openmpi-bin openmpi-common
```

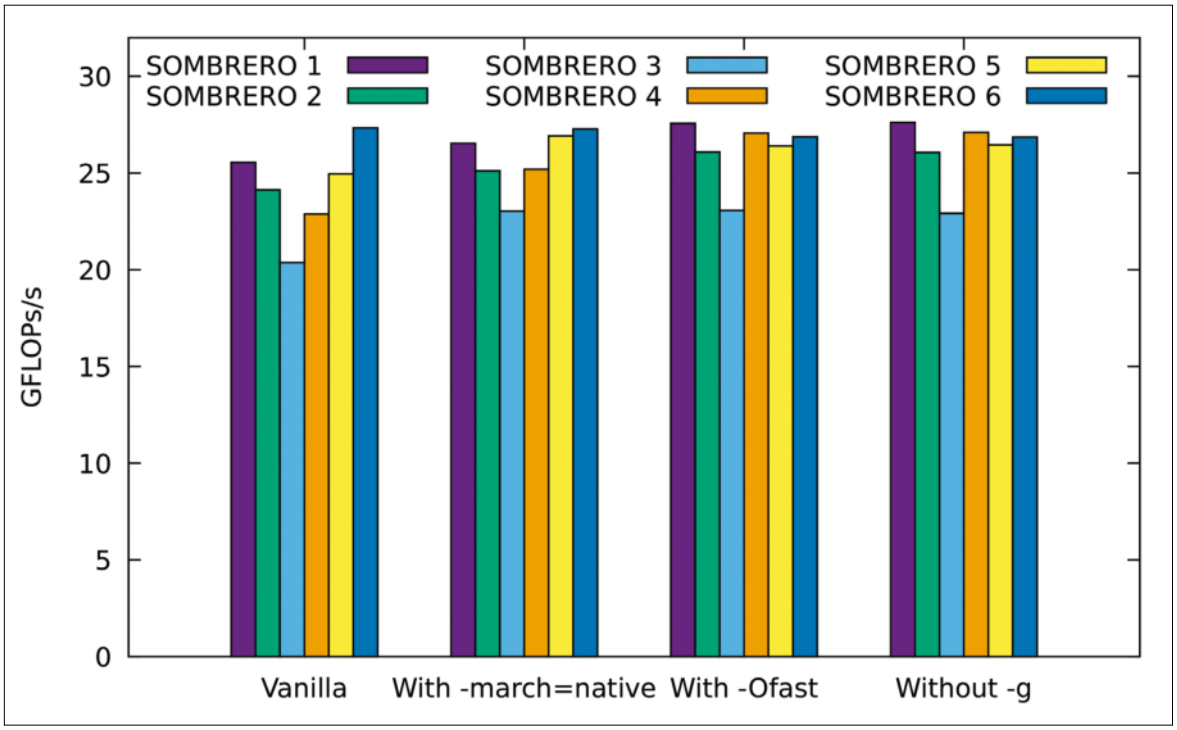
Since you've installed *Git*, you'll be able to update to the latest version of *Sombrero* from the repository at **https://github.com/sa2c/sombrero**:

```
$ cd ~/sombrero
$ git pull
```

The default settings are enough to get you up and running, so type `make` and *Sombrero* will build itself. It includes six different benchmarks, each from a different formulation that theoretical physicists are studying to see whether or not it could describe reality, so while *Sombrero* builds you'll see the same files compiled six different times, once for each benchmark. Also included is a shell script that runs each benchmark in turn, and gives a summary of the results. Since *Sombrero* is a highly parallel benchmark suite designed to run on some of the largest computers in the world, you need to tell it that you want to run it on a single CPU core, and to use the smallest problem size it knows about. To do this, use:

```
$ ./sombrero.sh -n 1 -s small
```

You'll immediately see some output from the first benchmark about the setup – since this test does not communicate between processes at all, the 'operations per byte' is infinity. Within a couple of minutes you'll see the results of each benchmark start to appear. The headline figure is the second `[RESULT]` line within each block—the one ending `Gflops/seconds`. 'Gflops' here is



On this laptop, -march=native gives a noticeable bump to almost all benchmarks, -Ofast gives a small speedup to some, and removing -g has little effect.

short for gigaflops, or 'billion floating point operations' (where a 'floating point operation' is any operation such as addition or multiplication done on a number with a decimal point). Since most supercomputers are used to crunch numbers for science and engineering applications, this gives a pretty good comparator between different machines.

Get parallel

So far this is only using a single core of your CPU. Since you'll be hard-pressed to find a computer with only one core in 2019 (*Jonni's Eee PC–Ed*) a good next step is to run using every core in your machine, to get the highest number you can. To find out how many cores your machine has, you can check the specs, or you can look in **/proc/cpuinfo**. (If your CPU has hyper-threading, you'll need to divide by two, since *Open MPI* doesn't believe that those are real cores.) Put the number you get in place of the `1` in the above command to run on all cores. For example, for a six-core workstation, run:

```
$ ./sombrero.sh -n 6 -s small
```

It's worth noting that because of the particular problem *Sombrero* is based on, it can only test a number of cores that is divisible only by 2 and 3. For example, 6 (2 × 3), 24 (2 × 2 × 2 × 3), or 32 (2 × 2 × 2 × 2 × 2) will work, but 10 (2 × 5) or 14 (2 × 7) won't work, and will give you an error.

Now you've successfully brought your entire machine to a halt, how about doing the same with your entire network? If you have more than one Linux machine joined by a network, you can try running *Sombrero* across every machine you have access to, forming a makeshift Beowulf cluster. First off, follow the instructions up to now for every computer that you want to participate. This will check that *Sombrero* works on each machine individually, and also set up the executable so that *Open MPI* can find it on each machine. Next, you need to configure SSH so you can log in from one machine to another without a password.

To activate the SSH server, run

```
$ sudo systemctl start ssh
```

On Red Hat-derived distributions, this will be `sshd` instead of `ssh`. Then on each machine in turn, generate an SSH key by running

```
$ ssh-keygen
```

and pressing Return three times. Once this is done, you can copy the key to your other computers with

```
$ ssh-copy-id destination_machine
```

where `destination_machine` is the hostname or IP address of the machine to copy to. When prompted, enter your password for the destination machine to let the copy happen. If you have two computers you'll need

QUICK TIP

If you're running on a Raspberry Pi, even the 'small' test is too big to fit into the system's 1GB of RAM. You can specify a custom size: try -l 16x12x12x12.

» WHAT IS A SUPERCOMPUTER?

A modern supercomputer isn't too unlike a home or office network. A bunch of Linux servers (or nodes), often using Intel processors and mounted in racks, can all operate independently, but sit on a common network. In the fastest supercomputers, each server has one or more GPUs installed to perform the heavy grunt of the computation. Red Hat Linux is the most common distribution used, with Debian-derived distributions being more of a rarity.

While a typical Ethernet connection may be used for managing the nodes and logging into them, they are also connected together by an ultra high-speed interconnect such as Infiniband or Omni-Path. The nodes need to work cooperatively on large problems, and don't want to sit idle waiting for data to arrive, so planning the network layout is a crucial part of designing a supercomputer. While most machines will use a relatively familiar hierarchy of switches, some machines targeting specific applications will design the network around that, giving exotic shapes like dragonflies and six-dimensional toruses!

Intel CPUs will be of the high-end server varieties, since this gives a much higher number of cores per node: current-generation chips give up to 24 cores on a single chip, compared to eight on the highest-end Core i9. GPUs will be computationally-focused NVIDIA Tesla cards – often with no display output at all.

QUICK TIP

If you don't have more than one machine to play with, you can make a comparison plot of any parameters you can tune. One particularly useful plot is to compare different compiler options, to see which is fastest.

to do this once on each, if you have three it will be twice on each, and so on. This is because MPI on each machine needs to be able to talk to every other machine, and you won't be able to type in the password once the program is running.

Now you'll need to let *Open MPI* know what machines you want it to use. To do this, create a text file called **hostfile.txt**, and in it write down the hostname of each computer you want to use. Repeat the name as many times as you want processes to run on that machine; for example, if you want your dual-core laptop called `pygmy` to join in a computation with a six-node workstation called `pyxis`, **hostfile.txt** will look like:

```
pyxis
pyxis
pyxis
pyxis
pyxis
pyxis
pygmy
pygmy
```

With the setup all done, you can test your new cluster with the command

```
$ ./sombrero.sh -H hostfile.txt -s small
```

In an ideal world, this will be faster than the score for one computer. That's not guaranteed, though – that depends on how fast your network is, and how fast the other computers are. Since the individual steps of the benchmark have to happen in lock-step across the entire cluster, the progress will be dictated by the slowest machine. Because of this, removing a particularly slow computer from the mix may enable things to speed up.

This is a problem that occurs frequently in 'heterogenous computing'. For many problems, load-balancing techniques can be used to adjust the amount of work each machine does based on its capability; however, most benchmarks don't bother with this, both because the underlying problems don't lend themselves to it, and because the systems that will be tested generally have very uniform components.

» TESTING AT SCALE

Since supercomputers rely heavily on their high-speed network to let computations be distributed across many nodes, it's important to test how well this works. There are two ways this can be done: weak scaling tests, where the problem size grows with the number of nodes used, and strong scaling tests, where the same problem is divided into smaller and smaller chunks. *Sombrero* supports both kinds of tests, defaulting to strong scaling, since that is the harder problem.

If you have enough computers available, you can run a strong scaling test. First, set up a set of node lists, called **nodelist1.txt**, **nodelist2.txt** and so on, using multiples of 2 and 3.

```
$ for NODES in 1 2 3 4 6 8; do
  ./sombrero.sh -H hostfile${NODES}.txt -s small | awk \
  'BEGIN {printf("'${NODES}'");}
  /^\[RESULT\] Case.*Gflops.seconds/ {printf("\t" $4);}
  END {printf("\n");}' >> strong.dat
 done
```

Then, to plot the results:

```
gnuplot> set style data linespoints
gnuplot> plot for [i=1:6] 'strong.dat' \
using 1:(column(i+1)) title 'SOMBRERO '.i
```

The ideal case is a straight diagonal line; in reality, the more nodes are used, the further from straight the line gets, as more and more communication has to happen.

Pursue the need for speed

So far, you've been running *Sombrero* with the default compiler settings, which are `-std=c99 -g -O3`. Now, those are fine settings to use for testing, but won't get the highest possible performance numbers for your system. To get this, you need to give the compiler more directions on what you want it to do to get the best performance. You do this by editing the **Make/MkFlags** file, which defines variables to be read by the compiler. Opening this file in your favourite text editor, you'll see:

```
#Compiler
CC = mpicc
CFLAGS = -std=c99 -g -O3
LDFLAGS =
```

Since we installed *GCC* earlier, you might expect to see `CC = gcc`; instead, `mpicc` is used. `mpicc` is a wrapper around *GCC* that *Open MPI* provides, and makes sure that all the correct communications libraries are linked.

To start with, the `-g` flag enables debugger symbols. This is useful when you're debugging problems with software, but can in some cases slow down performance slightly. Remove this flag now, and run `make` again. Re-running *Sombrero* now, either on your local machine or across a cluster, should reveal a slight speed increase compared to the previous run. The next step is to tell the compiler what kind of processor you're going to run on.

By default, *GCC* only allows itself to use processor features that are on a generic x86-64 CPU – this limits it to features from processors released in 2003. CPUs have come a long way since then, so enable *GCC* to use these new features by adding the flag `-march=native`. If you want to run on a different machine than the one on which you're compiling, you can pass other options than `native` to the architecture parameter; the *GCC* manual has the full story. Re-compile with `make` and re-test now, and you should see a more significant increase in performance.

GCC has hundreds of knobs you can tune to try and extract the best possible performance from your machine; you can browse the full list at **https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html**. This might seem like cheating, but it's an important part of both benchmarking and running software on large machines. The compiler is designed for a wide range of software and hardware, so sometimes needs help to know how best to make your software run. On a typical desktop, the performance difference is small enough that compiling every piece of software from scratch and testing every option will spend more time than it saves. But when the software you're working with is running on hundreds or thousands of identical machines for weeks or months at a time, you want to make sure that you're not wasting even five per cent of the available power.

In more extreme cases, a hardware manufacturer will rewrite all or part of a benchmark to be faster on their hardware. Again, while this seems unfair, in some cases

it's necessary to let the benchmark run at all. For example, *Sombrero* currently can't run on GPUs; if a manufacturer wanted to show off the power of their GPU-based supercomputers, they would need to adapt it so that it could run there. This is allowed, provided that the problem being solved doesn't change; that would be a different benchmark entirely.

Plotting the results

What good is a set of benchmark data without a set of colourful graphs to show off? While you could copy and paste numbers into *LibreOffice Calc* by hand, it makes a lot more sense to use a tool specifically designed for text-based data, so let's get started with *gnuplot*.

For this, the first thing you'll need to do is transform the results logs into a more tabular format. You can do this using *awk*. To start with the previous example:

```
$ ./sombrero.sh -H hostfile.txt -s small | awk \
'BEGIN {printf("pyxis and pygmy");}
/^\[RESULT\] Case.*Gflops.seconds/ {printf("\t" $4);}
END {printf("\n");}' >> results.dat
```

This will append one line to the end of **results.dat**, with seven columns separated by tabs. The first will contain the label `pyxis and pygmy`, with the remaining six containing the results of the six benchmarks. You can repeat the process for various configurations by editing **hostfile.txt** – perhaps pyxis by itself and pygmy by itself for comparison. Change the value of the `label` variable so that each line gets a unique label.

Now that your results file contains a table of results, you're ready to generate a plot. Start *gnuplot* with:

```
$ gnuplot
```

This puts you into the *gnuplot* interpreter, with a prompt that looks like `gnuplot>`. Tell *gnuplot* that you're going to plot a grouped bar chart, with filled bars and a black border:

```
gnuplot> set style data histogram
gnuplot> set style histogram cluster gap 1
gnuplot> set style fill solid border rgb "black"
```

Next, ask it to automatically scale the horizontal axis, but make sure that the vertical one starts at zero. Bar charts starting away from zero are generally used as the tool of those who want to mislead you about the data they're presenting!

```
gnuplot> set auto x
gnuplot> set yrange [0:*]
```

Then make sure that *gnuplot* knows that the data file uses tabs as its separator, since by default it will use any white space to separate values:

```
gnuplot> set datafile separator "\t"
```

Graphs aren't particularly useful without axis labels to tell you what the information means. Add one with:

```
gnuplot> set ylabel "GFLOPs/s"
```

Finally, you're ready to plot the data. This command makes use of a loop, so that you don't have to type the same commands over again for each column:

```
gnuplot> plot for [i=1:6] 'results.dat' \
using (column(i+1)):xtic(1) \
title 'SOMBRERO '.i
```

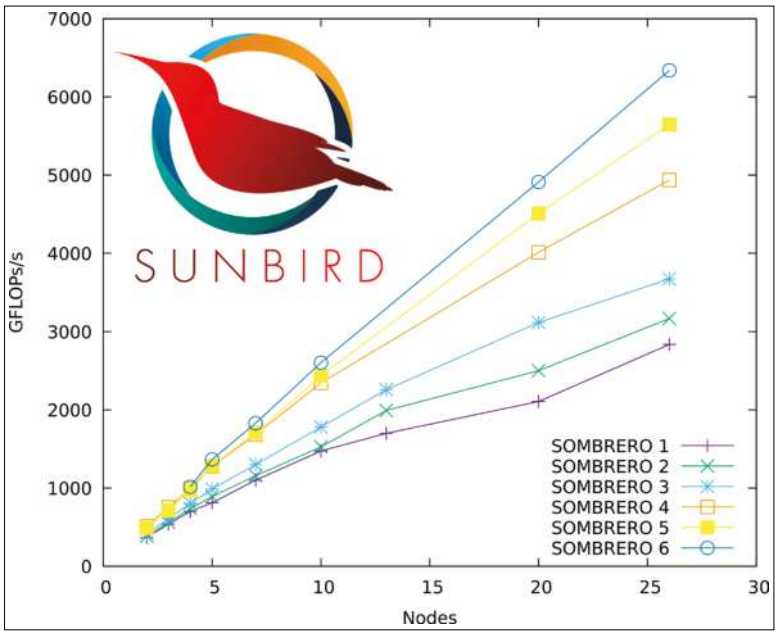
And with that, you're done! A plot appears in a new window, showing a labelled comparison of your different runs as a grouped bar chart. If you want to, you can export it as a PDF or image file, to email to your friends to show off, or print out and hang on your fridge.

In this article, we've explored how you can benchmark your system and even your entire network by using the *Sombrero* benchmark. So next time someone claims to have a faster setup than you, or you're trying to justify buying a new one because your machine has slowed down in its old age, you can back up your arguments with cold, hard data. LXF

» TOP TRUMPS FOR SUPERCOMPUTERS

Once upon a time, the *Linpack* benchmark – used to calculate the Top500 (**www.top500.org**) – was the only benchmark in town for ranking supercomputers. However, since that benchmark is designed to do as many floating point operations (FLOPs) and as little else as possible, it doesn't always paint a good picture of how well a machine will perform when it tries to do something more useful. Hardware manufacturers want to sell more hardware, and they sell hardware by having the biggest numbers, so they design machines to do better at *Linpack* than anyone else. Because all hardware design involves compromise, this can have the perverse effect of harming the machine's performance in properly interesting problems, just to get a better TOP500 score.

Because of this, a whole zoo of benchmarks and rankings has started to spring up. For example, Green500 measures supercomputers' energy efficiency when running *Linpack*: crucial, when as a rule of thumb the electricity costs for running a supercomputer for its lifetime are about the same as the cost of buying the machine. Graph 500 (**http://graph500.org**) focuses on data-intensive applications rather than the computationally intensive *Linpack*. *Sombrero* focuses on a mix of raw computation, speed of reading from RAM, and time it takes to send data across the high-speed network between nodes – with the balance between these shifting between the different benchmarks. For each benchmark, a different supercomputer will reign champion.



The Sunbird supercomputer at Swansea University shows good strong scaling when running Sombrero. The high-numbered tests give near-ideal scaling.

AUDACITY

Advanced audio recording and editing

John Knight returns to his old recording friend Audacity once again, and finds things have become a little more advanced.

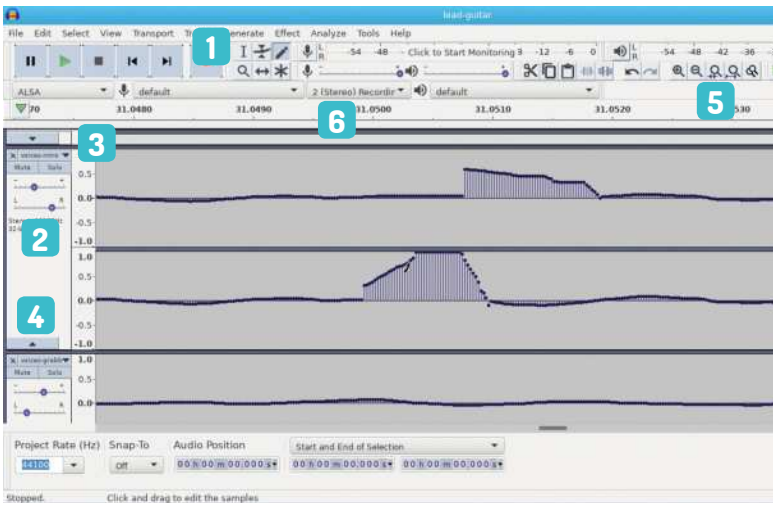
OUR EXPERT

John Knight writes ebooks on how to play the drums, when he's not playing with a Commodore 64 emulator.

Big audio suites are great, but they can be intimidating to say the least – sometimes you just want something that works without having to think too much about it. Enter *Audacity*, from: **www.audacityteam.org**.

Audacity has become popular on all desktop platforms, and has started popping up on YouTube tutorials quite a bit. Known for its ease of use and intuitive interface, *Audacity* is perhaps the audio equivalent of *Microsoft Paint* (*erm...–Ed*) – it can't do deeply complicated tasks, but then it was never intended to do so. It's designed around simple controls that most people can guess their way around... though that doesn't mean you can't achieve some great results!

Audacity's interface



1 Main Toolbar
Where your primary tools are kept, such as Selection, Time Shift, Zoom and Envelope.

2 Track Controls
Tools for muting, soloing tracks and changing the gain and the pan. This is where you really shape your soundscape.

3 Track Menu
Provides a bundle of more advanced controls, such as mono-stereo conversion, as well as renaming and moving tracks.

4 Collapse Button
Squishes up any track into a kind of sliver which can be expanded later.

5 Zoom Controls
Dedicated buttons for specific zoom functions, including zooming into a selected portion of a wave, or zooming out 100%.

6 Recording Options
Choose which audio system you want to use, which device to record from, mono or stereo, and change the playback device.

Although technically *Audacity* can record multi-channel digital audio, the interface really isn't designed for it – it's meant for simple recording in mono or stereo. If you're looking for a *Pro Tools* replacement, *Audacity* isn't it: try *Ardour* or *QTractor* instead. But if you're looking for an easy to use wave editor and a fast, simple way of making stereo mixes, it may be what you're looking for. *Audacity* lets you get deep into the wave form and perform easy cuts and edits, making it very popular among technicians who want to cut together quick edits, and apply basic effects without getting bogged down by a complex interface.

Nevertheless, *Audacity* has undergone some changes lately, and at first glance some long-time users might have their nose put out of joint. So for the new users we'll give you a walkthrough of the major features, and for the veterans we'll show you what's changed and how to adapt to it. We won't be showing you all the features, but we will cover enough of the essentials to start a workflow.

Sounds good

In order to show you how *Audacity* works, we first need to have some audio to play with. You can either import some existing audio or just record random sound (blank silence is fine) – it doesn't really matter, as long as there's an audio track of some sort.

If you want to add some existing audio tracks, you can run through a maze of GUI prompts if you really like, or you can just click and drag them onto the editing field. As for recording audio, just hit the Record button in the main toolbar, and if you're using an internet mic, clap your hands a few times – that should show up on the waveform. Press Stop when you're done and we'll be able to look at *Audacity* properly.

The Device Toolbar lets you change which sound card you want to use, and which audio host will run the system – likely ALSA, maybe with the option for JACK. For your recording device, you will probably have options for Default (most likely an ALSA device) or Pulse. ALSA will probably run quicker and be kinder on the system, though Pulse will likely be easier to get working if your system is already configured around it.

Perhaps of chief importance is the number of Recording Channels. Here you can choose between

one channel for mono or two for stereo. Don't just choose stereo because you think stereo = better. If you're recording a single instrument, it may be easier in the long run to record this in mono, even if you will ultimately be creating something in a stereo image. If you're recording something like an FM broadcast or capturing computer audio, this is something you will want to do in stereo.

Audacity's toolbar – which has the snappy name of '*Audacity* Tools Toolbar' – houses the main tools like Select, Draw, Time Shift and Zoom. By default the Selection Tool will be enabled, and is used for highlighting audio before doing things like cutting and pasting, muting sections and so on. Cutting and pasting is done by the usual shortcuts, so you'll have no problems there, and you can use the Delete key on selected bits of audio. You can select the magnifying glass for Zoom: left-clicking zooms in, and right-clicking zooms out. You can zoom right down to the grains of each waveform if you want to do some hardcore manual editing, and if you select the Draw tool, you can even re-draw the shape of the wave.

If one of your tracks is out of sync with the other recorded tracks, you can use the Time Shift tool to correct it. If you click and drag left or right it will move the timing forwards or backwards, and you can even move a track to before the zero-second mark.

Further on the right are the zoom controls, all with different magnifying glass icons, which replicate the Zoom tool. The icons with the + and - let you zoom all the way in to the fine points of a waveform, or back out so that you can see the track as a whole.

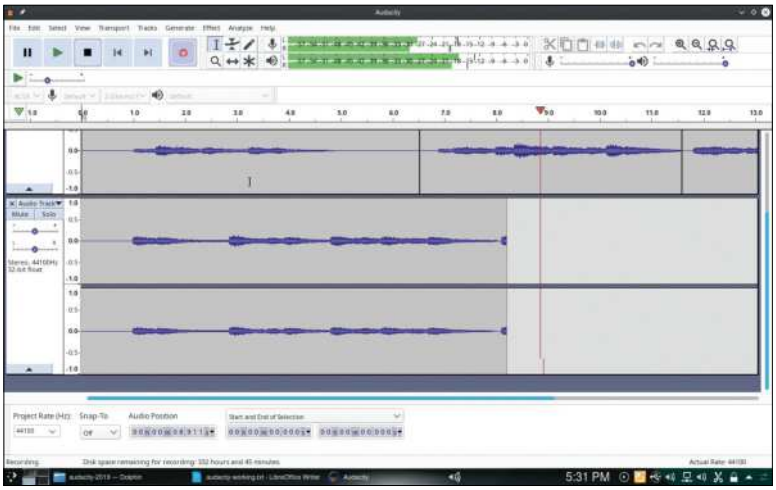
Going solo

Less obvious, but crucial when you get the hang of things, are the next two controls: 'Fit selection in window' and 'Fit project in window'. When you have some audio selected, 'Fit selection in window' will zoom in until your highlighted selection fits perfectly on the screen – no more and no less. When you need to see the project as a whole again, 'Fit project in window' zooms all the way back out in one click.

Moving to the tracks in the main editing field, track controls are on the left. Each track has a close button which removes the track, and a drop-down menu. Plenty of advanced options are in the drop-down menu, but the main controls you'll use are the Mute and Solo buttons, plus the two sliders. The Mute and Solo buttons should be fairly self-explanatory, but remember that you mute each track individually, rather than all of them en masse. While the Solo button will obviously mute all other tracks, you can 'solo' more than one track – something that will annoy many pedants!

Moving down to the two sliders, the first is the Gain control. In simple terms it basically turns up and down the volume of each track; in audiophile terms, this is where pre-amp adjustments happen. The second slider is the Pan control, which lets you move sound left and right in a stereo image.

Looking to the advanced controls in the drop-down menu, the first option is Name, where you can rename each track individually. The Move Track Up and Move Track Down features will become essential when you start doing elaborate multi-layered editing, as this lets you re-order the tracks.



Audacity is still the easiest way to do multi-track editing and recording, even though it's now slightly more complicated.

More advanced users will want to check out the options Swap Stereo Channels, Split Stereo Track and Split Stereo to Mono. These features are so much quicker and easier than what you encounter on professional editing suites, and are perfect for anything that involves switching between mono and stereo.

Get recording

Getting started with the first track is the hardest part of any recording session. You've got to make sure the correct input is selected, get the levels right, and check that your recording is actually coming through on your program's waveform. If you're very lucky you'll be able to just hit the Record button and it will work. But even with something as easy as *Audacity*, you'll probably be greeted with a blank wave and need to spend ten minutes switching between inputs to find what's wrong.

Before we get too negative, try your luck and hit the big Record button. Make some noise and if all is well, you will see the sound outputting into a waveform. If you were greeted with silence and a flat line, first make sure the recording option for that channel is enabled in your mixer and that it's not muted. Otherwise, try changing entries from the Recording Device drop-down.

Unless you have a Goliath of a sound card, there shouldn't be too many entries to test, and you can probably choose between several channels under

QUICK TIP

Do you have some pesky audio that only runs through one speaker? You can fix it by clicking the track's drop-down box and choosing Split Stereo to Mono.

» ADDING EFFECTS

Because *Audacity* cuts out the complex audio buses that underlie big editing suites, adding effects is incredibly easy. Just select whatever audio you want to apply an effect to and choose it from the Effect menu – and if more input is required, most effects have a settings window with a handy Preview button. As for the effects included by default, you will want to pay particular attention to Fade In and Fade Out, Amplify, and Equalisation. Note that Amplify is useful for making things both louder and quieter, as you can simply enter a negative value if you want to turn something down.

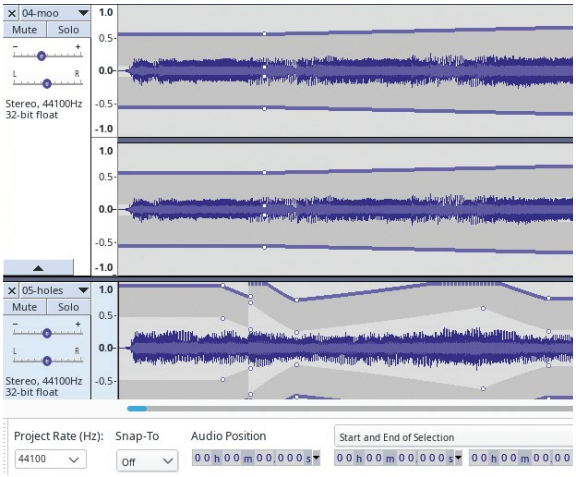
While *Audacity* has a number of effects pre-installed, we recommend installing extra plug-ins. *Audacity* supports LADSPA, LV2, Nyquist, VAMP and VST plug-ins, and you should be able to find quite a few plug-ins in your package manager. Just browse around and install anything you like the look of! Anything to do with reverb, chorus or flange is probably a good idea.

QUICK TIP

A cool feature veterans may not know about is the Wave Color entry in the drop-down menu, which has four different colours, and is meant for easily differentiating between instruments.

Default or Pulse. Default will probably use ALSA, which will run much faster, but if you're using Pulse audio, open the properties for your sound card. In KDE we had to open the Advanced tab in the Audio Volume section, and from the Profile drop-down box, we used Analogue Stereo Duplex. If you're using digital speakers, just try anything that enables an analogue input. Once you have something that works, write down what you did and don't mess with it – record everything you can while it's working! After this, *Audacity* should be smooth sailing. Now to tweak the recording levels.

The smaller the wave in height, the worse the signal-to-noise ratio will be, making the track hissy and probably muffled. The louder the recorded input, the more likely it will 'clip' the edge of the waveform and sound horrible and distorted. You want to strike a balance where a good amount of space in the wave form is used, but leave room in the dynamic range for sudden loud notes, such as a big cymbal crash. Keep turning up your instrument, input or mixer desk until your waveform is getting near the edge, and back it off if it starts clipping. Depending on your underlying audio system, you may be able to turn the recording levels back down with the slider in the top-right of the window.



The Envelope tool lets you make smooth and continual transitions to volume over time, such as a long fade-out.

Running tracks

Recording a piece of audio is great, but you can do that on just about anything. What makes *Audacity* so popular is that it does multi-track layered recording, and probably makes it easier than any other program out there. If you're new to the concept of multi-track recording, it allows you to record sounds one at a time in layers, slowly building up a soundscape.

For instance, the most common usage is in music where you may record each musician in isolation, rather than recording the band as a whole. You may lay down a drum track first, and with that first drum track in their headphones, the bassist can record their track in a new second layer. Then the guitarist could hear the drums and bass playing through their headphones and record their guitar in a third layer. With the instrumentals finished, the singer can record their vocals.

If you're a seasoned *Audacity* user, you need to know that things have changed. When you hit the Record button, *Audacity* used to create a new track automatically and start recording in that. Now when you click Record, *Audacity* will start recording on the end of the current track, and just keep on adding to it.

If you would rather record into a new track, use the keyboard instead of the mouse. Pressing R records onto the end of the current track, but Shift+R records into a new track. If you're determined to have things the way they used to be, you can change it by choosing Edit > Preferences from the main menu. Open the Recording section and in the Options field, click 'Always record on a new track'.

Note that *Audacity* now has latency correction built-in, but some audio device settings may result in an error regarding latency timing. If this is the case, you can also tweak your latency correction settings in the Preferences window. In the Devices section, you can tweak the buffer length and 'Track shift after record' settings in the Latency field.

Track-on-track action

Not only can *Audacity* do multi-track recording, but you can also edit each recorded layer individually, making tweaks to any layer without upsetting the others. Going back to the band example, if you simply recorded them in one live image, you would have to mix them together as a whole and you wouldn't be able to make many adjustments. But with separate tracks you can correct mistakes and tweak each recording individually.

For instance, if the guitarist plays a bum note or the drummer coughs during silence, you can simply mute their second of bad audio and the rest of the song will be unaffected. Or, if one instrument is too loud, you can turn the volume down on just that track without affecting the volume levels of any other musician.

If you want to turn a track up or down, just move that track's Gain slider. If you wish to mute a second of audio, simply highlight the portion of audio you don't want and press Ctrl+L, or click Edit > Remove Special > Silence Audio. For more advanced users who want to do volume automation, there is the Envelope Tool in the main toolbar. This lets you make smooth volume

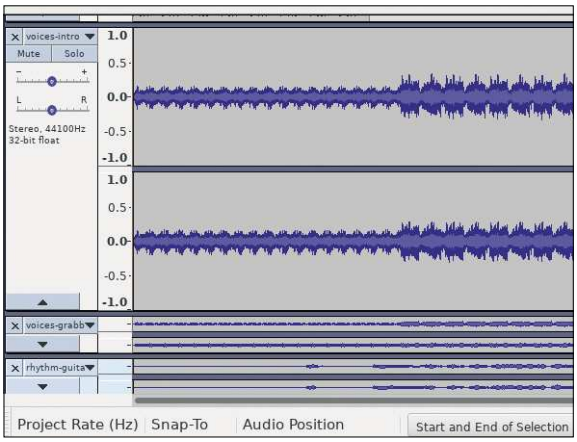
» RECORDING WITH JACK

Running *Audacity* with JACK rather defeats the purpose of its simple interface, but maybe there's some kind of effect chain you want to create, or some kind of MIDI program you need to use, and the only way to do it is JACK. First, you need to make sure JACK runs properly. We recommend installing *QjackCtl*, which provides a straightforward GUI for starting and configuring JACK.

If you try to start JACK and get an error, first try specifying your soundcard by clicking Setup, and in the Settings tab will be the Interface drop-down menu. Change the setting from '(default)' to your soundcard's actual device name. If this doesn't work, make sure Driver is set to 'alsa' instead of 'dummy', and if it still doesn't work, try disabling 'Realtime'.

To change *Audacity* from using ALSA to using JACK, open your system settings by choosing Edit > Preferences and in the Devices tab will be the Interface section. For Host, open the drop-down menu and change the entry from ALSA to 'JACK Audio Connection Kit'.

It's best to leave everything at default settings and see if you can record as-is. If you need more JACK connectivity, you change *Audacity*'s Playback and Recording devices to JACK, though you may need to manually connect ports in *QjackCtl*'s Connect window. JACK is messy and temperamental at the best of times. Don't use it with *Audacity* unless you really need to!



Each track has a handy Collapse function, which is perfect for focusing on one track when the number of tracks becomes overwhelming

transitions by creating two lines: one on the edge of the track and one on the edge of the waveform, which can be dragged around to reshape the wave's amplitude. The outer line will bring down volume overall, but the inner line can be dragged upwards to increase the wave's amplitude.

The really fun bit of *Audacity* is creating custom stereo images. You can take a large selection of tracks, and by sliding the Pan control on each, you can create a large stereo image in your own virtual soundscape, making your music or home movies really come alive.

If this is something you'll do a lot of, a common mixing technique is to actually record individual voices or instruments separately in mono. This makes it much easier to manage when you want to create a stereo mix, because if the imported track is already in stereo, it adds unnecessary complexity and filesize. If you've recorded each voice in mono, you simply import all of the tracks into the same session and assign their place in the stereo mix with the Pan slider.

A common scenario is with digitally recorded drums, where each drum mic will have recorded to a separate channel, with each channel outputting to a separate wave file. If you import all of these files at once you will just hear a dull mono image, but if you use the Pan slider, you can place each drum and cymbal virtually based on where they would sit in real life.

For instance, overhead mics will usually be mixed far left and far right, the hi-hat on the left, the ride cymbal on the right, and the kick and snare will usually be mixed around the centre. If you put the snare slightly left and the kick slightly right, you will be able to hear each drum more clearly in the mix and avoid overlapping frequencies.

Trade-free exports

Simply saving your work isn't going to be of much use, because the only thing that will read *Audacity* files is *Audacity* – you need to export your work. If you click File and look at the Export sub-menu, you will see a number of options. Firstly, we need to differentiate between the 'Export as...' options and 'Export Selected Audio'. The 'Export Selected Audio' option is simple enough: whatever you highlight with the Selection Tool is saved to an external audio file. The 'Export as...' and 'Export Audio' options save an image of your work exactly as it sits now – mixer settings, effects, and everything. This is something you'll want to do when you've got everything just the way you want it.

WAV files are something of a de facto standard in the mixing world because pretty much anything can read them, although no one's going to shout at you for using FLAC files. Use WAV or FLAC wherever you can while you're mixing – most compressed formats like MP3 or OGG are lossy and should only be used for distribution after you've already made a proper master. If you don't like WAV, OGG or MP3, more encoding options are available using the Export Audio function.

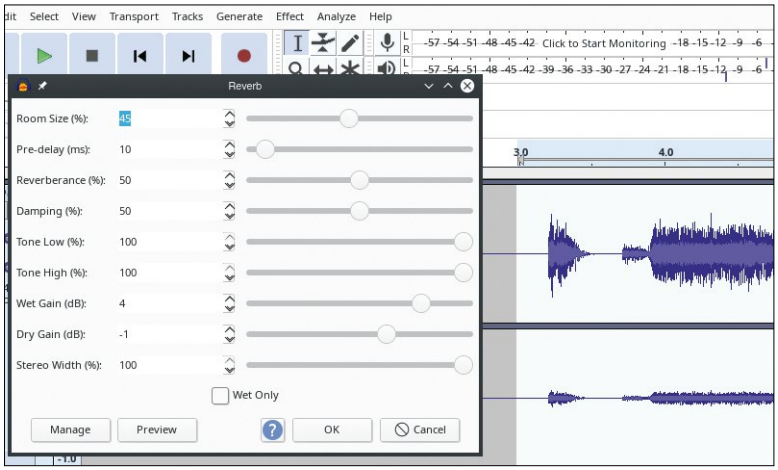
For *Audacity* veterans, there is some cool new time-saving functionality for the Tracks menu, under the sub-menu Mix. There are three options: Mix Stereo Down to Mono, Mix and Render, and Mix and Render to New Track. The first option is pretty self-explanatory, converting any selected stereo tracks into mono. Mix and Render converts all your current tracks into one single track, including any volume or pan adjustments, added effects and so on. Note that you will have to 'Select All' to make this work properly.

Lastly, Mix and Render to New Track does the same Mix and Render, except it leaves the existing tracks as they are and creates a new rendered track beneath your current tracks.

All mixed up

Audacity has undergone quite a few changes of late, most of which push it toward being a more viable and professional multi-track recording program. However, these changes also add to its bulk and complexity; the simple truth is that *Audacity* is no longer as simple as it once was, which may scare off some new users and even put off some veterans.

Nevertheless, *Audacity* is still the friendly and intuitive program it has always been. As long as it doesn't gain much more in complexity, existing users should be able to adjust and *Audacity* will maintain its niche as the easiest way to get into multi-track recording and editing. LXF



By simplifying and stripping back the underlying system, applying audio effects is much easier than it is in big editing suites.

QUICK TIP

To make a split in the audio, choose the Selection tool, click where you want the split, and click Edit > Clip Boundaries > Split. Now you can move sections of the wave independently.

LINUX MUSIC PRODUCTION

John Knight jumps into the music making scene, looking at the best apps for writing tunes from scratch to full mixing production.

Linux sound editing has grown from a quirky alternative to a genuinely viable platform with respect and commercial support. Each year, more commercial audio products are released on Linux, as more companies start taking it seriously.

But fear not. There is no need to buy – or even pirate – expensive commercial editing suites when all you need to create music is just waiting to be downloaded. As the proprietary commercial scene grows, the free software audio scene is growing even faster. With each year that passes, Linux music-making is becoming stronger and richer, with more quality applications than we can possibly cover.

Entire distributions and variants such as Ubuntu Studio, KX Studio and AV Linux exist to create dedicated music-making machines, with low latency kernels to optimise recording performance, and software bundles pre-configured to make recording as easy as possible. You can choose to run one of these specialised bundles or stick with any good mainstream distro.

Regardless of the quality on offer, the hardest thing about changing between any production suite is establishing a new workflow, so we'll show you how to hit the ground running and get past any initial barriers that trap new users.

Obviously we have a limited amount of space, and though we would love to include everything, we have decided to go with the most universal and well-supported software packages that are available with the most distributions. Along the way we will explore how to get your audio system up and running, the best tools for musical composition, a lazy way to record demos, and examine two of the best digital audio workstations for creating full productions.

Whether you're a hardened studio veteran with roomfuls of equipment, or just an angry teenager with a cheap laptop, there should be something for everyone – even those without any musical instruments! There's everything you could need to build your songs from the first few test drum raps on your desk to a finish production, and finally break away from those pesky proprietary workstations.

Image credit: Leonard Ritter, aka "paniq"

The master jacker

Before we can start making music, first we need to get the tricky configuration bit out of the way: that thing called JACK.

The JACK Audio Connection Kit is a sound server intended for low-latency recording, allowing extraordinary levels of interconnectivity between audio and MIDI, and some truly elaborate sound rigs. Although most of the applications here can run without JACK, it's really the foundation behind most advanced Linux audio. Unfortunately, it usually takes a bit of tweaking to get started with it.

The most popular tool to configure JACK is *QjackCtl*. You'll find this in all the main repos (Ubuntu, Fedora et al), so `apt install qjackctl` and it'll also pull in JACK as a dependency. Once installed, open *QjackCtl* and press Start. If you're lucky JACK will start the first time, but even then, you'll probably need to make a few tweaks.

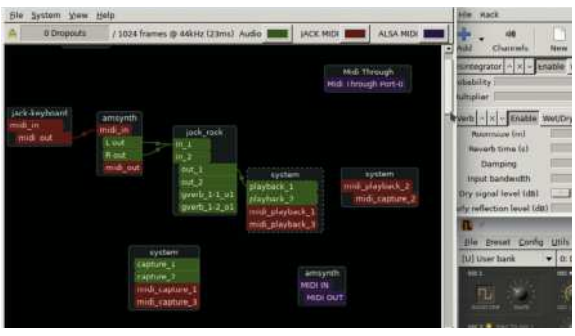
If JACK (*see feature* **LXF191**) won't start, the first thing to try is to ensure that 'Driver:' is set to 'alsa', and that 'Interface:' is set to your actual sound card, rather than just being left on '(default)'. If you want to use MIDI, the MIDI driver will probably be set to 'none', so you'll need to choose 'seq' from the menu. Next, try adding your user to the audio group, and if you're still getting no joy, you may have to disable 'Realtime'.

JACK might also fail to start if you're already using something that needs audio and won't relinquish your sound card's permissions – you may need to log out if so. It's best to start with a clean session, and don't run anything extraneous, especially not web browsers.

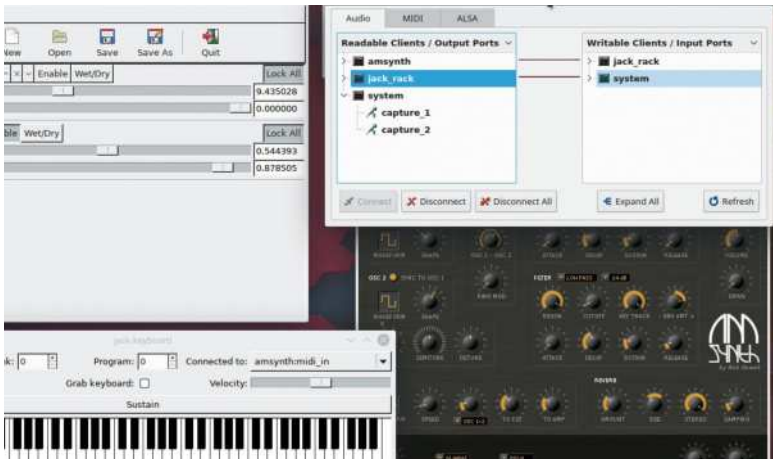
Note that when you finally have JACK running, your sound card will be taken over entirely, so anything outside of JACK's ecosystem won't work. However, we're not done yet: there is still latency to contend with.

Latency is a problem that plagues all sound engineers, regardless of software or equipment they use. If there is a substantial delay between a new track and the track(s) it is recording over, you have latency issues. Don't just turn the latency down, though – the lower the latency, the greater the load on the system, and the more likely there will be weird clicks and pops.

You need to find a sweet spot: start with the defaults, and gradually adjust the latency until you can remove any delay without overburdening the system. Once JACK is running you can start anything that's JACK-reliant.



It's tempting to click everywhere until something works, but be patient and methodical – you need to be able to replicate your connections.



Setting up JACK's connections can be quite a faff, but thankfully you can save your configurations for future usage.

Making connections

Making the correct connections in JACK is probably the hardest part of the process. Start by clicking the Connect button in *QjackCtl*.

The Connections window is divided into two halves: Readable Clients/Output Ports on the left, and Writable Clients/Input Ports on the right. It is also broken into three tabs: Audio, for sound using JACK; MIDI; and ALSA, for Linux's underlying sound system. Much like the cables involved with real musical equipment, programs or modules can be assigned as outputs and connected to the inputs of other programs by means of virtual leads.

To make a connection, click the output client on the left, then the client on the right you want to connect to. For any connection JACK deems possible the Connect button will become available, and when clicked, a line will be drawn from left to right, indicating a virtual lead has been connected.

In the image (*see above*), we are using multiple JACK clients/modules to create a new sound. In the MIDI tab, 'jack-keyboard' is attached to the MIDI module 'amsynth'. Back in the Audio tab in our image, 'amsynth' is attached to effects program JACK Rack, which is applying kooky modifications to otherwise boring synth sounds. Finally, JACK Rack is connectedto 'system', which is the final audio output.

If *QjackCtl*'s Connections window doesn't float your boat, *Patchage* (*see left*) has a completely different interface. Each client is a box you can drag around in space, and clicking between compatible ports will make a connection. It's certainly more elaborate, but feels more like the strewn cables one finds on the floor of a studio practice session!

Tools for composition

No matter how you might prefer to create songs, there's a bit of software for you.

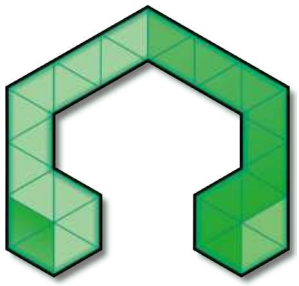


Image credit: Tobias Doerffel (CC BY-SA 3.0)

The app called *Linux Multi Media System* (*LMMS*) has become the daddy of loop-based music making and easy MIDI editing. The important bit of *LMMS* is the sidebar on the left, which has samples, file management and instrument plug-ins – and instrument plug-ins are what really power *LMMS*. Included are a bunch of weird instruments and different synthesisers, but we'll be focusing on the component called Sf2 Player, which runs MIDI soundfonts, enabling you to compose MIDI-based music without external hardware.

» AUDACITY

www.audacityteam.org

Over the page we'll be looking at fully fledged Digital Audio Workstations (DAWs), and these programs can be daunting to say the least! If you just want to do some basic multi-track recording (that is, layering one take over another) without the complexity of a full DAW, *Audacity* is your best bet.

Although it's technically possible to do it, *Audacity* isn't really suited to multi-channel digital recording, such as digital drum recording with lots of mics, as the interface is really geared around mono and stereo.

However, it's brilliant for home demos and has become something of a de facto standard for simple audio editing as it is intuitive enough for most people to navigate, while being powerful enough to do some serious wave editing.

That said, *Audacity* has become more complex lately, and it's more likely you'll need to troubleshoot something than in times past. For example, if you want to record a new separate track, press Shift+R instead of the Record button, as now *Audacity* keeps writing to the end of the last track you recorded.

If *Audacity* in its current form isn't to your taste, it's worth trying a build from a couple of years ago instead. For a proper tutorial to the program, see tutorials in **LXF213, LXF 228** and **LXF249**.

Audacity is great for taking down easy demos when you're still writing and can't be bothered with a big scary DAW!

If you drag Sf2 Player onto the Song-Editor window it will create a new track. Click the track's Sf2 Player button and a new plug-in window appears, with a piano and sound controls. If you click the piano now, no sound will come out of it – first you need to load a soundfont. In the middle of the window are two menus: FILE is for choosing the soundfont, and PATCH is for choosing which instrument to use (and there may be many to choose from). Once you've loaded your soundfont, clicking a piano key should now make a noise, and your computer keyboard will act as a substitute piano.

The Song-Editor has a collection of nodes for activating notes or samples: left-clicking enables a node, middle-clicking disables it and double-clicking opens the Piano-Roll to create new tunes.

In the Piano-Roll you can either record notes on the fly, or draw them by hand. Clicking on the piano keys on the left previews each note, and as long as the Piano-Roll is in Draw mode (click the pencil icon if it isn't), clicking in the blank field creates a note that looks like a solid bar.

If the note is the wrong length, you can click and drag the right of the note to make it longer or shorter. If you've played the wrong note, click and drag the note up or down the piano to adjust it, or move it into another octave. To delete a note, just right-click.

If drawing out riffs isn't your thing, you can use your computer keyboard to play notes in real time, and if you press the record button, *LMMS* will transcribe the notes for you on the fly. The beauty of *LMMS* is that you don't have to choose between one mode or the other – you can combine both methods: drawing notes by hand, or playing them live. If you fudge a note while recording, you can redraw it later.

Whether you're into electronic music or not, *LMMS* is an excellent way to navigate songs and sequence musical arrangements. *LMMS* may not be a recording suite, but for those happy enough using synthesised instruments and virtual drumkits, there's genuinely all you need to construct a song without having to buy extra hardware.

Hydrogen

http://hydrogen-music.org

Unlike looping programs, *Hydrogen* is made by drummers for drummers, and is a great way to experiment with beat-making – even for people who have never held drumsticks. If you want to compose easily charted percussion arrangements across an entire song, *Hydrogen* is the way to go.

Perhaps the easiest place to start is the bottom half of the screen, with the Pattern Editor. This is where you write drum bars, and the numbers along the top indicate which beat your note sits on. On the left is a list

Image credit: Roland Geider

LMMS has a brilliant method for composing MIDI, plus some crazy instruments including a classic Gameboy and even a Commodore 64.

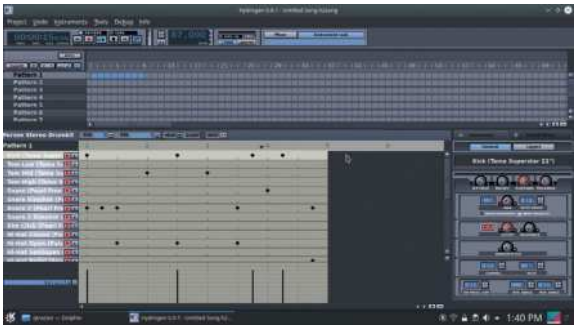
of every instrument within the drumkit, and if you want to preview a sound, just click an instrument.

The Pattern Editor is pretty straightforward, but something that may throw you off is the Size function. Size is set to 8 by default, and the default 'resolution' is eighth notes (quavers), so the Size function will be set to eight eighth notes to make a full 4/4 bar. That's slightly confusing at first, but if you want to make a pattern in 5/4, the Size should be 10, and if you want something in 7/8, the Size would be 7 – and so on.

If eighth notes won't suffice, and if you need smaller subdivisions, you can change the Resolution to 16th notes (semi-quavers), or go right down to 32nd or even 64th notes. If you need something in a triplet-based subdivision (for blues or jazz, for example), you can choose the same numbers, but with a T – such as 8T for eighth-note triplets.

Moving onto the Song Editor in the top half of the screen, there is a list of patterns on the left. Click a different pattern from the list to open another pattern below in the Pattern Editor; it's a good idea to name each pattern – just right-click and open Properties to change it.

On the right is a giant field with empty squares. Each square represents a bar of music: clicking each square will sequence that bar when the song reaches that point in time. For anyone into highly complex percussion arrangements, multiple patterns can run together.



Hydrogen gives minute control for intricate drum arrangements, and can be extended with JACK for amazing drum sounds.

If you don't like the default drumkit (GMKit, which admittedly is pretty mediocre), there should be a selection of alternative drumkits available as a package in your repositories. To the right of *Hydrogen*'s screen is the Sound library tab. This lists whichever kits are installed; to load one, right-click on the title of the drumkit and click Load.

Once you've finished your drum arrangements, you can export your project as an audio file to be used in a recording suite later – speaking of which, it's time to look at these...

» PLUG-INS, MIDI AND SOUNDFONTS

While your package manager is open we recommend installing just about every audio plug-in available (search for LADSPA, DSSI, VS, and LV2) – and you should install jack-keyboard if you plan on recording MIDI without an external device.

Every plug-in you install will augment your system's functionality. While we would normally be selective about what to install, plug-ins aren't massive, so unless disk space is at a premium you may as well install each one you can find!

Note that MIDI by itself doesn't have any sound; MIDI is merely a set of standardised instructions for what sounds to play and when. Therefore, if you try to run raw MIDI on your system, you will be greeted with silence. MIDI-based hardware such as keyboard synthesisers use internal recordings of instruments captured at specific pitches to create a whole sound spectrum.

In the modern audio world we use soundfonts: audio files of recorded instruments that can be used by your sound card, played according to MIDI instructions. Some soundfonts are tiny, whereas others can be elaborate recordings of multiple gigabytes.

However, SoundFont (note the capitalisation) is a commercial term for a file format that is supported in hardware by products like Creative's Sound Blaster range. Popular software synthesisers like *FluidSynth* bypass the need for such dedicated hardware, allowing you to run general soundfonts through any sound card. If you would prefer to avoid soundfonts entirely, *Yoshimi* comes with its own set of sounds. For a list of free soundfonts and plug-ins, see: **https://lmms.io/documentation/Useful_resources**.

Digital audio workstations

What's the word when it comes to ultimate sound editing? DAW...

The digital audio workstation is to music editing what *Photoshop* is to graphic editing: an all-singing, all-dancing piece of software for recording, editing and mixing all sorts of audio. Traditionally DAWs have been complex, expensive and decidedly proprietary – think things like *Pro Tools* – but there are great Linux alternatives.

Ardour is an open source, collaborative effort of a worldwide team. Over the past few years it has has undergone quite a few changes, so we'll be giving that primary focus. We'll also look at *Qtractor* very briefly – if you would like to know more, see LXF242 for a proper rundown.

Although *Ardour* has plenty of MIDI capabilities, its MIDI editing prowess isn't highly regarded, so we'll just focus on audio editing for *Ardour*, and leave MIDI and some other areas to *Qtractor*.

Image credit: Thorsten Wilms (CC BY-SA 3.0)

Ardour https://ardour.org

The 5.x series of *Ardour* has a lot of new features, including Windows support, the ability to control group faders with one controller, Lua scripting, a new plug-in wiring system and at last, built-in plug-ins for basic effects.

The GUI has undergone many improvements, with a tabbed interface instead of many small windows, and a far more intuitive workflow. It appears you can now run *Ardour* through ALSA, without the need for JACK – this should greatly simplify usage, though presumably this also limits your mixing functionality.

During setup, you're asked where to place *Ardour*'s recorded files; whether to use a hardware mixer or have *Ardour* play material as it's recorded; and whether to use the master bus directly, or use an additional Monitor bus (just use the default if you don't know). Next, you're given a choice of session templates for getting you straight into a workflow. However, in order to show you the GUI in a pure, simple form, choose Empty Template for now.

Lastly, you will need to define the options for Audio/MIDI Setup. Our version defaulted to ALSA, though we had to specify the right input and output devices. Once started, *Ardour* will take over your sound card entirely, so don't expect to multitask between something like *Ardour* and online cat videos.

If you prefer JACK, you will be given a series of configuration options. If you don't really know what you're doing here, just make sure the right device is chosen and click Start. If JACK has already been configured, *Ardour* will start it automatically for you.

Once inside the main screen you will see a track called Master. This is the Master bus: everything passes through it, but you can ignore it for now – first you need to create a track you can record over. To make your first track, you can either do it the long way by clicking Session > 'Add Track, Bus or VCA', or you can just right-click on the track panel to the left of the screen.

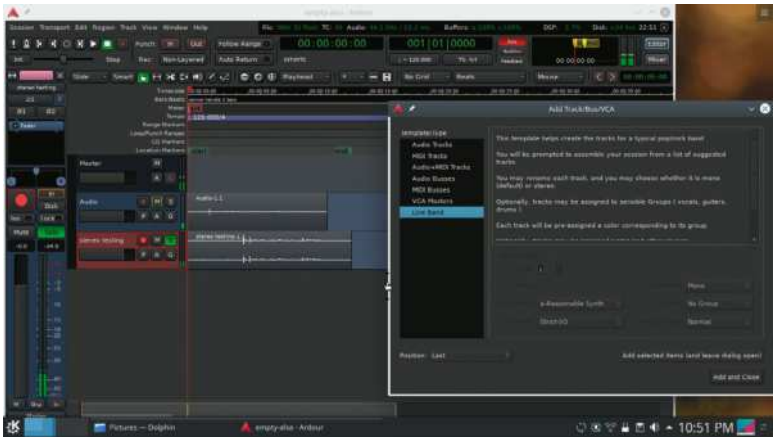
When the Add Track window appears, there are a selection of track types to choose from. We're just concentrating on audio, so stick with the default choice of Audio Tracks. In Configuration, choose whether you want mono or stereo, give your track a name and click 'Add and Close'.

Back in the main editing screen, look to the track panel on the left and click the Record button on the new track. From the main toolbar, click *Ardour*'s primary Record button (which will flash, meaning it's ready to write), and click the Play button to start recording.

What a mix up!

Before we go any further, we recommend turning on View > Show Editor Mixer. We won't cover the complex issue of audio buses here, but we can show you how to apply simple mixing and effects.

Each track has some basic controls on the left, such as Mute, Solo, Gain and record with this track. If you right-click on the track controls there are more options, including basic things such as track colour, and more



After many years of awkwardness, Ardour now has a smooth interface that may woo lazy Audacity users to its side.

» ADDING EFFECTS IN ARDOUR

Changing the Pan is, let's say, unintuitive, so we'll start there. By default the Pan slider is locked, and instead you control the width of the stereo image. To unlock the Pan slider, right-click the Pan control and choose Stereo Balance from the drop-down menu.

To add an effect, select the track you want to enhance, and click the Solo button to hear the track in isolation. In the mixer panel, right-click the blank area of space under 'Fader'. From the drop-down menu, select New Plugin, where you can browse all the plug-ins you've installed. When activated, a window will open with your plug-in's settings.

Back in the mixer, your plug-in will have a little green light, indicating it is enabled. If audio is playing, you can enable and disable the plug-in to compare the difference in sound.

Hardened veterans are likely to appreciate Ardour's improved bus design compared to earlier versions.

advanced options like track alignment and automation. However, to make further mixing adjustments like changing Pan, you'll need to open the mixer, the button for which is in the top-right corner.

Exporting audio is fairly straightforward in Audacity 5.x. From the main menu choose Session > Export > Export to Audio File(s). It's best to leave these options alone if you don't know what you're doing; with the defaults it should save your work as a snapshot of the entire session. A warning: *Ardour* crashed the first time we tried to export, so click Save first!

Ardour has been around for 14 years now and is synonymous with Linux audio – there is even a modified proprietary variant, *Harrison Mixbus*, for serious audiophiles who want a big name behind their software. Nevertheless, it has always suffered from something of an awkward interface. Now it seems the tables have turned, and rival projects will need to up their game.

Qtractor http://qtractor.org

Although *Qtractor* has evolved into a fully fledged DAW, it was originally meant to be just a MIDI sequencer with some DAW features. *Qtractor*'s strange evolution has resulted in an endearing hodgepodge of all the major editors, with bits of *Rosegarden*, *Ardour*, old *Pro Tools*, and chunks of *Cakewalk* and *Cubase* for good measure. The last time we used *Qtractor* it was prone to crashing, but developer Rui Nuno Capela has put a lot of time into fixing bugs.

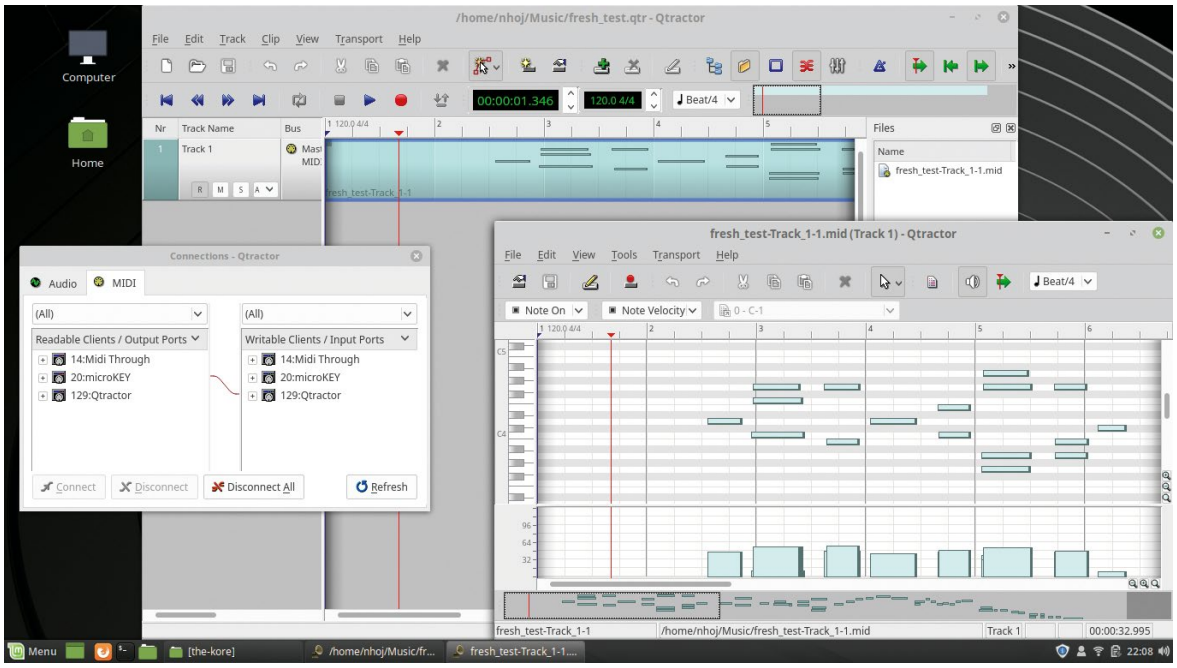
Unfortunately, starting a project involves running through a series of prompts that may be intimidating and somewhat confusing (much like the bad old days of *Ardour*!). However, once you're up and running, *Qtractor* has some killer editing features.

Rather than deleting chunks of audio, you can grab the edge of either end of a track and just drag it inwards. Rather brilliantly, you can also drag it back to 'uncrop' as it were, even past the start of recently pasted bits of audio. Interestingly, if you hold Ctrl when you click and drag a clip's border, *Qtractor* will generate more of the same audio based on what's gone before it. If you have your BPM set correctly the Snap feature can break down your music into perfect bar-length chunks, which can be dragged around at will. And don't worry if your wave spills over the edge of what you've grabbed, as you can always 'uncrop' it.

If you're into MIDI, *Qtractor* has a similar piano roll editor to *LMMS*; double-click on a piece of recorded MIDI to open it. If you use a synth plug-in when adding the MIDI track, you can avoid involving JACK. *Calf Fluidsynth* is particularly popular and can load new soundfonts, though *Yoshimi* if you can't be bothered!

Exporting audio was very confusing last time we covered *Qtractor*, but it seems to have a reasonable GUI flow now. To export the session as a whole, choose Track > Export tracks – Audio, and follow your nose. If you choose Master as your output, it will export everything into a wave file, regardless of whether it's audio or MIDI.

Compared to *Ardour* nowadays, *Qtractor*'s cluttered interface is quite confusing. But if you can get into a flow, and understand the *Qtractor* way of doing things, the combination of wave audio and MIDI may be all you ever need to make music. LXF



Qtractor's interface may be messier than Ardour's, but the MIDI functions are more powerful and the clip editing is utter genius.

Image credit: Andrew Fitzsimon

EMBEDDED COMPUTING

Getting hands-on with embedded hardware

Mike Bedford delves into the world of embedded computing, and discovers that a small amount of hardware can certainly go a long way.

OUR EXPERT

Mike Bedford continuously finds himself being drawn back to embedded computing.

QUICK TIP

If you enjoyed learning about embedded computing with the Elegoo UNO Project Basic Starter Kit and want to learn more, you might be interested to know that the same company offers a more advanced kit that could teach you so much more. It's called the Elegoo UNO Project Super Starter Kit and it costs about twice the price of the basic kit.

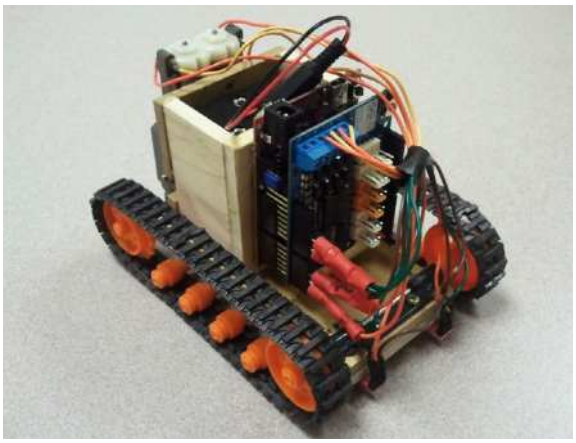
Single board computers (SBCs), typically credit-card sized or smaller, are now a familiar part of the computing scene, thanks in no small part to the Raspberry Pi family of products. The Pi has spawned lookalikes from several companies, which have also attracted their supporters, but perhaps less well-known are those SBCs intended exclusively for embedded applications. Our aim here is to introduce this category of SBCs, and to illustrate how you could put them to work using a starter kit that you can pick up for just a few pounds.

To start, though, just in case this is a new concept to you let's describe what we mean by an embedded application. This is one that works in the background in such a way that the user might not even be aware that they're using a computer, in the broadest sense of the word. So, for example, household appliances such as microwave ovens or washing machines are generally controlled by embedded software.

If you've delved into the Raspberry Pi, you'll no doubte be aware that it has several GPIO (General Purpose Input Output) pins. These can be used to interface to external components such as pushbuttons, LEDs, motors, sensors and such like, so they could be used to control a robot or a whole range of other real-world devices. In this sense the Pi could be considered as a platform for embedded computing, but it's quite different from those SBCs that are intended exclusively for embedded use.

The Pi is a general-purpose SBC; in fact many people use it to learn to code, and completely ignore its interfacing capabilities. This dual nature has one important consequence: Raspberry Pi computers run an operating system, commonly some flavour of Linux. By way of contrast, dedicated embedded SBCs generally don't run an operating system.

The lack of an operating system might seem to be a serious drawback, but for embedded applications it's an asset. If your use of an SBC is just to learn about embedded computing, the Pi approach works well, but if you want to put your SBC to work in a real application, an operating system is an overhead you can do without. Perhaps the major drawback with an operating system is the time it takes to boot. We can imagine that a car wouldn't be too popular if you had to wait 30 seconds for Ubuntu to start up before you could drive off. It also has implications in the amount of hardware you need. Many embedded applications will work perfectly well with a much less powerful processor than one you'd need to run an operating system. It's not just the processor – commonly an embedded application won't need any graphics hardware, and often it won't need a host of interfaces such as Wi-Fi or Ethernet. Not only does this have cost benefits, it also reduces the power requirement which is a major advantage for battery-powered equipment.



Robotics is an excellent example of an embedded application, this one being based on Arduino hardware.

CREDIT: Marcmccomb, CC BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Line_follower.jpg

Introducing Arduino

Rarely will you find an off-the-shelf SBC inside a drone, a TV remote control unit or a food processor. Instead, in order to minimise the cost of these sorts of high-volume products and to reduce size, the computing hardware will usually be incorporated into a custom-designed circuit board, alongside all the other electronic circuitry. For low-volume DIY applications, though, and especially for learning about embedded computing without being hampered by an operating system, a dedicated embedded SBC makes a lot of sense. This is where Arduino boards come to the fore.

Raspberry Pis and similar SBCs tend to be built around an Arm processor of some type. By way of contrast, many of the Arduino boards are based on

the ATmega328 from Microchip. There's a lot more than just a name that separates them, though, as a few figures will reveal. The processors on Pi-type boards have a 32-bit of 64-bit architecture, they generally have two, four or eight cores, and they're usually clocked somewhere between 1GHz and 1.8GHz. The ATmega328 has an 8-bit architecture, a single core, and the clock speed is often just 16MHz.

The difference in memory and storage is just as stark. Instead of capacities measured in hundreds of megabytes or gigabytes, Arduino boards will commonly have 32KB of flash storage for the program, and user storage of a few kilobytes of RAM and a similar amount of non-volatile EEPROM. For embedded applications with no need to support an operating system, you might be surprised at how much you can do with such an apparently modest specification.

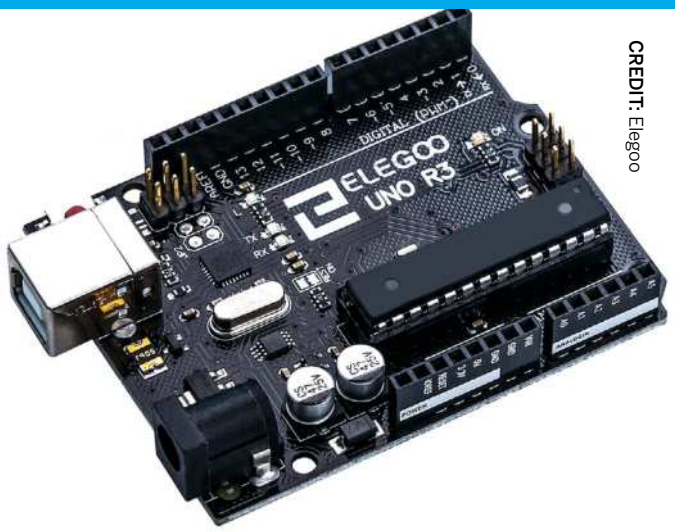
Finally, before moving onto something a bit more practical, we need to spell out what we mean when we talk about an Arduino board. Arduino was developed at the Ivrea Interaction Design Institute in Italy for educational use. Official Arduino boards – of which there are several – are available at **www.arduino.cc**. The design is open source and several companies have produced compatible designs. These should not be called Arduino boards, because the name has been registered, although third parties can refer to them as Arduino-compatible: several use names ending in 'duino' or similar to make the point.

Generally, third-party designs are cheaper than those sold by Arduino. There are also boards on the market that appear identical to Arduino products, but use the Arduino name and logo illegally – these should be avoided. Oh, and just a quick word on what we mean by the phrase 'Arduino-compatible': the bottom line is that different manufacturers might use it to mean different things, so be sure to read up carefully if compatibility is important. At the software level, it means that it will work with the Arduino IDE – that's the Integrated Development Environment that you'll use on a separate PC to develop code before uploading it to an Arduino. Most compatibles offer this level of support.

At a hardware level, the most common understanding of compatibility is that the I/O pins are compatible so that the board can be used with Arduino shields: add-on boards which are roughly equivalent to HATs in the world of Raspberry Pis. Most so-called compatibles also offer this level of compatibility, but there have been reports of a few that don't.

The Arduino IDE

Because Arduino-type boards don't run an operating system, nor are they easily able to interface to a keyboard, mouse and display, software is developed on a separate PC and is then downloaded to the Arduino. The easiest way to do that is to use the Arduino IDE which is available for all major operating systems, but note that it runs under Java, so you'll have to install that if it's not already present on your PC. As with many IDEs, the one for Arduino provides all the tools necessary for code development. In particular, it includes a text editor in which you'll create and modify the source code, a compiler and linker, and a means of downloading the compiled code to the Arduino board. Because all third-party Arduino-compatible boards work with the official IDE, it's not included as part of the Elegoo UNO Basic Starter Kit that we're going to be using, so you'll need to install this IDE separately.



CREDIT: Elegoo

Arduino-based hardware, like this UNO board from Elegoo, is so much simpler than Raspberry Pis and similar SBCs, but it's ideal for controlling real-world devices.

Many of the repositories only have an outdated version of the Arduino IDE, so you'd be better off downloading the latest version from **www.arduino.cc/en/main/software**. There's also an online version that runs in a browser and saves your projects in the cloud. If you decide to use the online version, you will still need to install the Arduino Web Editor plug-in on your PC to handle the uploading of code to the Arduino. This cloud utility is worth considering because the IDE is just one element of Arduino Create, which is described as "an online platform that enables developers to write code, access tutorials, configure boards, and share projects". It also means that you'll always be using the latest version of the IDE. We decided to concentrate on using a locally installed IDE and we got it to work with no difficulty at all. Others report problems, so it's good to

QUICK TIP

For the ultimate in bare-bones embedded computing, forget about an SBC and use a single chip – for example from the PIC range of micro-controllers. Some of these cost just 60p.

» CODE DEBUGGING

We have some good news and some bad news. The good news is that software for many of the control applications you'll be writing – at least initially – won't be very complicated, so the likelihood of introducing programming errors is less than with many coding tasks. The bad news is that, if your software does go wrong, debugging code running on an Arduino SBC is tricky. In particular, the standard IDE does not provide facilities such as setting breakpoints, single-stepping, reading memory locations and so on.

What is possible, however, is to use the USB link between the SBC and your PC – the same link that's used for downloading code to the Arduino – to transmit data back to the IDE. This involves adding instructions to your code specifically for debugging purposes. Typically you'd use a `serial.write()` or `serial.print ()` instruction to output debugging data – such as the value of a memory location – to the serial port, having first initialised that port with `serial.begin()`. You are then able to make use of the IDE's serial monitor facility to inspect what the sketch has written.

Bear in mind that a mistake with the electronic circuitry attached to the board's input/output pins is often the problem. Many users report that hardware errors are more common than software errors.

QUICK TIP

We said that embedded computing platforms don't have operating systems, but at the top end they might. Devices such as set-top boxes, routers and other networking equipment, and medical electronics might well run an operating system. What's more, there's a good chance it could be Linux-based.

note that there's a forum where you can enlist community support. We should also point out that the first tutorial in the starter kit describes the installation of the IDE so, if you have problems, take a look at the PDF document that appears on the kit's CD in the folder for your language.

Elegoo UNO

We're now going to be showing you how to get some hands-on experience of embedded computing using the Elegoo UNO Basic Starter Kit. Elegoo sells directly from its website (**www.elegoo.com**) in US dollars, but it also sells via Amazon in several countries at much more competitive prices. At the time of writing, the Basic Starter Kit costs just £13.99 in the UK and $14.99 in the USA, from Amazon.

As the name suggests, the kit includes an SBC that's compatible with the Arduino UNO R3. We won't list the specification of that board; suffice to say that it's adequate for all the projects documented in the kit, and a whole lot more. The kit also includes a USB cable to connect the UNO to a PC, and a mini-CD containing a tutorial document and the code for all the projects, although they're also available online. The rest of the kit comprises several electronic components with the means of connecting them together and to the SBC.

The Integrated Development Environment enables you to develop code on a PC and upload it to an Arduino board when you've finished.

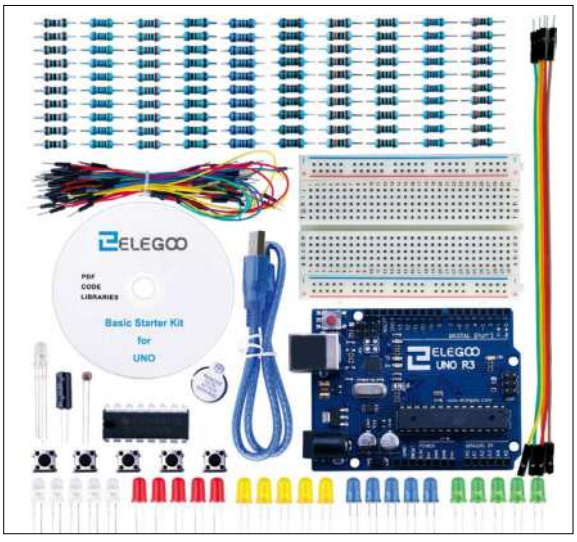
First is the breadboard on which you'll mount the various electronic components and make electrical connections between them. There are male-male patch leads for making connections between components on the breadboard, and for connecting the components on the breadboard to the input/output pins on the UNO, and there are also a few male-female patch leads. Turning to the components themselves, there are LEDs of various colours, pushbuttons, a buzzer, a photo detector and a tilt switch. There are also several components that are needed for interfacing, mostly resistors, including a 74HC595 IC – a serial-to-parallel converter that's used in some of the projects.

To see the UNO in action, just attach it to your PC using the USB cable supplied. Be careful not to put the board down on an electrically conducting surface, because this could cause shorts to the solder joints on the bottom of the UNO. As soon as it's connected to a PC, you'll notice that the green power LED illuminates and an orange LED starts flashing. The UNO is supplied with a simple program already flashed into memory, and it's this code which is responsible for flashing the LED, and serves to prove that the board is working properly. Now it's time to start up the IDE, which we assume you've already installed; this won't stop the LED from flashing. You're now ready to take your first steps in embedded programming.

Programming in an UNO

First we're going to see how to upload some code to the UNO and move on to modify that code. This mirrors the second lesson on the Elegoo UNO Basic Starter Kit CD, the first one being the installation of the IDE. This is an excellent place to start since it doesn't require you to wire any external components to the UNO's I/O pins.

Before doing anything, it's necessary to tell the IDE which Arduino board you're using, although it'll probably default to the UNO. To check, and to change it if necessary, go to Tools > Board. We're not going to create any new code to start, nor use one supplied in the starter kit, but instead we'll use some sample code supplied with the IDE. Open this at File > Examples > 01.Basic > Blink and it'll appear in the IDE's editing window. In Arduino-speak, the code is called a sketch.



CREDIT: Elegoo

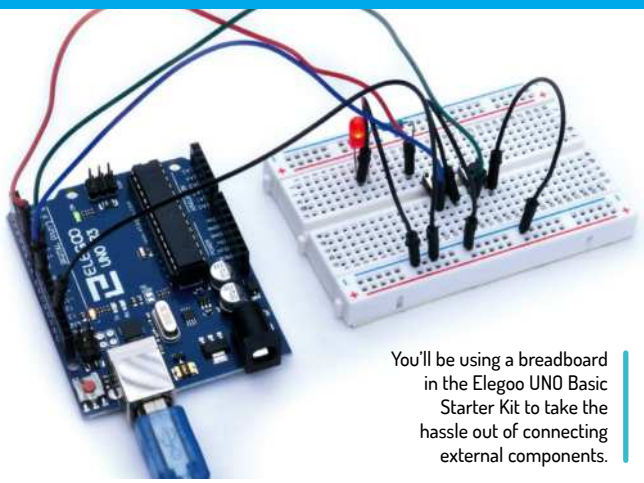
The Elegoo UNO Basic Starter Kit is a very cost-effective way of learning the principles of embedded computing.

» THE ARDUINO FAMILY

The Arduino family – even if we only consider the official boards – is much more diverse than the Raspberry Pi family, and becomes even more so when we include third-party products in the mix. The names don't really reflect where they are in the pecking order, though, so you'll need to consult a comparison table to decide which one to use.

Like SBCs that run Linux, Arduino boards are differentiated by the power of the processor and the amount of memory, but only to a degree. You might do better than the 16MHz 8-bit processor on the UNO, but you're not going to find anything comparable to the Raspberry Pi's processor, and only a few have more than 256KB program flash or 32KB RAM. For all but top-end embedded applications, you'd be more interested in the number of I/O pins and the size and form factor, and here Arduinos vary quite a bit.

You'll also find that some don't have headers for the I/O connections. While these are indispensible for learning and for development, or if you want to add a shield, for a real-world application much greater reliability will be achieved if you use one of the boards that just has 'pads' onto which you can solder the leads to external components.



You'll be using a breadboard in the Elegoo UNO Basic Starter Kit to take the hassle out of connecting external components.

The language is usually described as very similar to C/C++ with libraries for working with Arduino boards. Whatever you want to call it, though, so long as you have had some experience of coding, you'll soon get to grips with it. In fact, even if you're new to programming, with a bit of guidance in the form of online tutorials you should get the hang of it without too much trouble.

The purpose of this code is to flash the onboard orange LED, so it's pretty much the same as the code that was already flashed into the UNO. Even so, to check that everything's working, click the round icon with the right-pointing arrow in the toolbar. Watch in the blue status bar at the bottom and you should see 'Compiling sketch...' followed by 'Uploading...' and then 'Done uploading'. You might also have noticed that another couple of orange onboard LEDs flash briefly, indicating the transfer of data between the PC and the Arduino. Once you know the IDE is working, we'll change the behaviour of that flashing LED.

Towards the bottom of the sketch you'll find a line of code that turns the LED on, followed by one that waits for 1,000 milliseconds (one second), then one that turns it off again, and finally one that waits another second. These four instructions are inside a loop, so they execute forever. An easy change is to alter the lengths of the two waits from 1,000 to 500 milliseconds, so edit the sketch accordingly and upload the new code. After the other two LEDS have flashed, you'll notice that the orange onboard LED will start to flash more quickly. If you want to save your altered code, do it under a new name because the examples are read-only files.

Grip the hardware

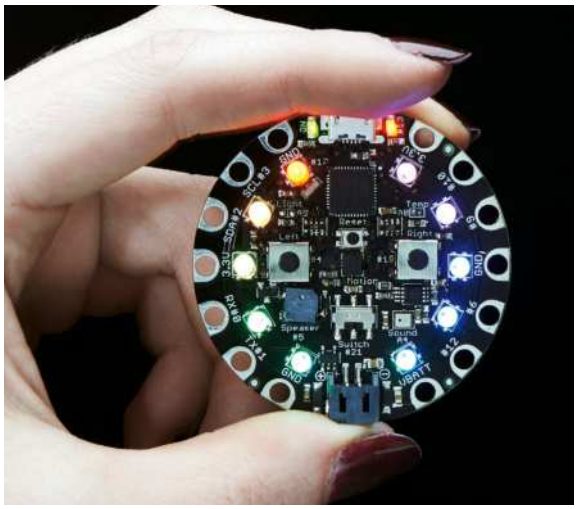
We're not going to go through all the tutorials in the starter kit documentation because that would be reinventing the wheel. The descriptions provided are excellent, giving both instructions and explanations, and you'd be well advised to go through all of them in sequence. Our aim here is just to whet your appetite in the hope and expectation that you'll get hold of this starter kit and, in so doing, get to grips with embedded computing. To do this, we're going to highlight just one tutorial, Lesson 5, which is a slightly expanded version of a classic in embedded computing. This enables an LED to be turned on with a pushbutton and turned off with another, so it illustrates the use of a digital input and a digital output. You can think of it as the equivalent of the 'Hello World' program, which is the first program many people write when learning to code.

In this example, we're going to use some hardware that's external to the UNO, which is the very essence of embedded computing. We're going to be mounting and connecting together a few electronic components on the piece of breadboard that forms part of the starter kit. Breadboard would never be used for a real-world piece of equipment because printed circuit boards are so much more compact and reliable. For learning, or for trying out ideas, though, they're ideal. Breadboards have an array of holes into which you can push the leads of components, and behind those holes are electrical connections so components can be wired together without soldering. This means that you don't have to buy a soldering iron, and components can be reused.

To understand the internal connections inside the breadboard, hold it with the red and blue lines running horizontally at the top and bottom. Behind the scenes, all the holes in the top row connect together, as do those in the second row, the bottom row and the row one up from the bottom – these are intended for the power supplies. The holes in the remainder of the breadboard are connected in vertical columns, but with a horizontal gap where you can see the gutter.

Now, add the various components to the breadboard as shown in the starter kit manual. You'll notice that, when you push a lead into a hole it will be 'grabbed', thereby holding it physically and connecting it to other holes internally. It doesn't matter if you use exactly the same holes as shown in the documentation so long as the components are positioned correctly with respect to each other. Next, load the code for Lesson 5 into the IDE from the starter kit's CD.

If everything's gone to plan, you'll find that the left-hand pushbutton turns the LED on and the right-hand one turns it off. If it doesn't work initially, try wiggling the component leads on the breadboard because it's all too easy make a poor connection when you add the add components to the board. Finally, before we leave you to work through the other lessons, how about trying your hand at coding by modifying the sketch? LXF

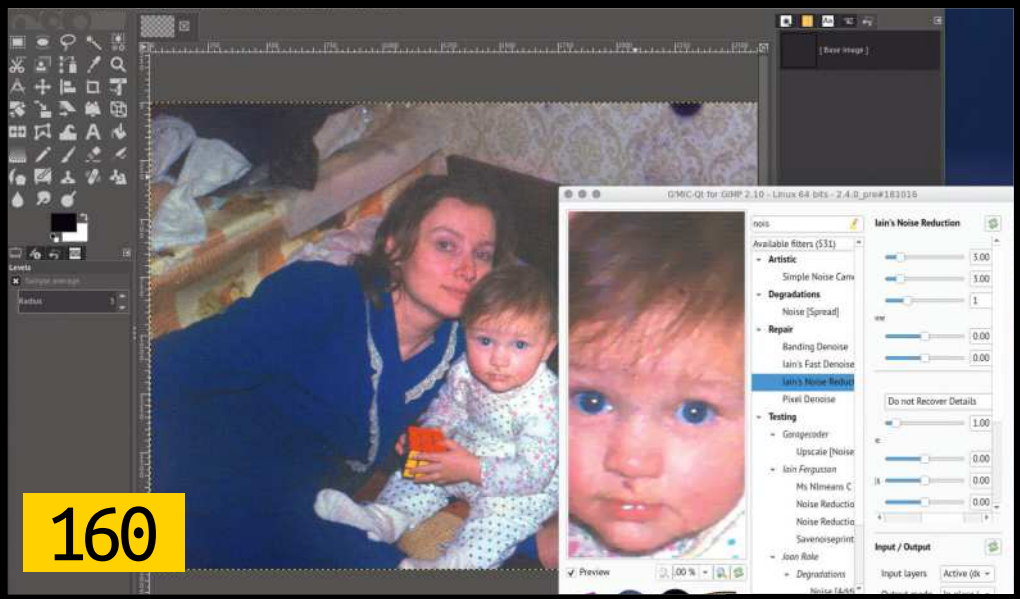


CREDIT: Adafruit

The Flora is an Arduino-compatible board from Adafruit that can be sewn into clothing, for wearable electronic applications.

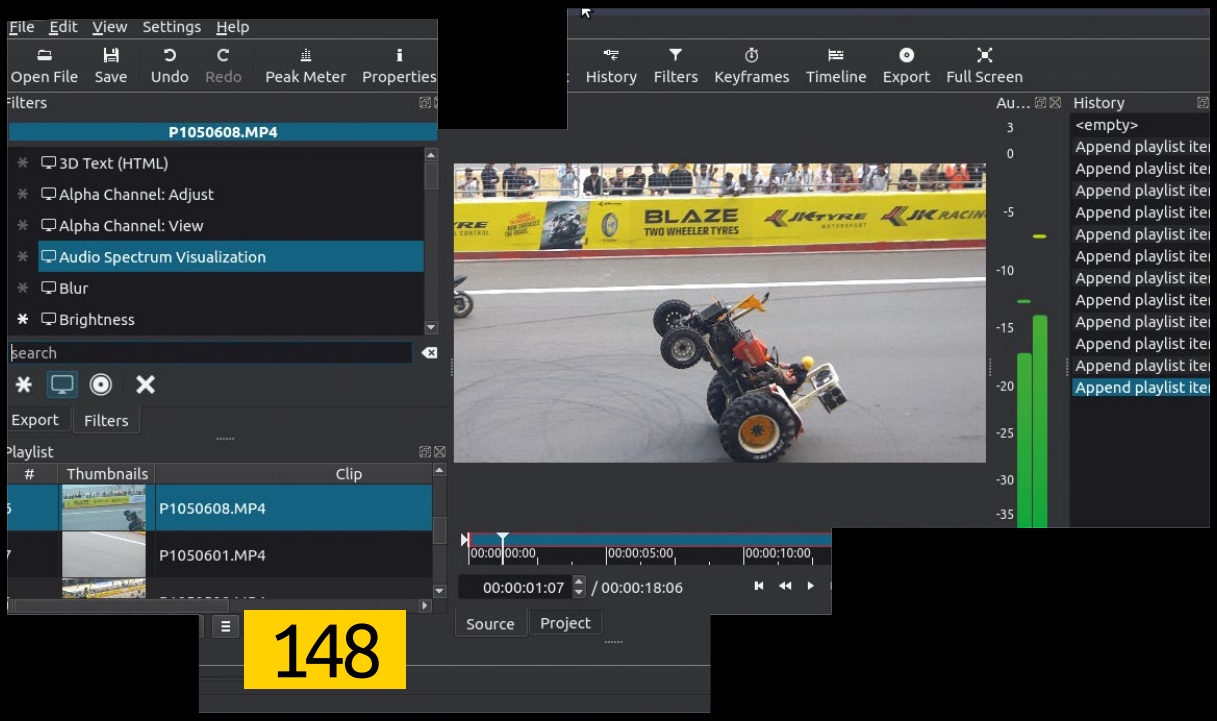
FOSS

126



160


166


File Edit View Settings Help
Open File Save Undo Redo Peak Meter Properties
Filters
P1050608.MP4
3D Text (HTML)
Alpha Channel: Adjust
Alpha Channel: View
Audio Spectrum Visualization
Blur
Brightness
search
Export Filters
Playlist
Thumbnails Clip
P1050608.MP4
P1050601.MP4
History Filters Keyframes Timeline Export Full Screen
Au... History
<empty>
Append playlist ite
BLAZE
TWO WHEELER TYRES
00:00:01:07 / 00:00:18:06
Source Project
148

FLATPAK

Get hold of up-to-date software with Flatpaks

Aaron Peters went all in on Ubuntu's .deb package format years ago, and has never looked back. That said, there's always time for a second opinion...

OUR EXPERT

Aaron Peters wishes he was a programmer, but sadly has to settle for telling programmers what to do in the form of software documentation.

Linux has made great strides in the area of software compatibility. Or, more accurately, software is available more consistently across the various distributions and their native package formats. Yet this still requires the hard work of packagers. They need to take an application and tweak it to fit the specific needs of different distros, and even different releases of the same one. However, a few new universal package formats have emerged, bringing the promise of "package once, run anywhere" (on Linux, anyway).

Flatpak (**https://flatpak.org**) is among the contenders Mayank Sharma reviewed in **LXF234**, and it's got a lot going for it. In this article we'll examine *Flatpak*, explore how to use it, and take a look at two important developments.

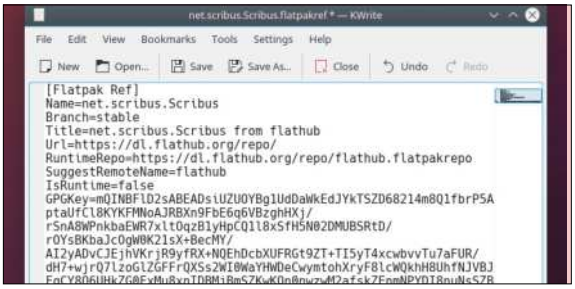
Mode of operation

Flatpak is both a package format, and a tool to install and manage those packages. Many Linux package formats work on the concept of dependencies, which means a developer can define that package A depends on package B. When a user tries to install my package A, the system will recognise the dependency and install package B along with it. This is how you `apt install` one application, and end up with a hundred packages to be downloaded and installed.

In contrast, *Flatpak* provides you both the application and everything it needs to run in a distro-independent format. This approach has its pros and cons:

Pros	Cons
Distribution agnostic	Uses extra space with duplicate files
Ease of installation	Installation by either root or non-root users
No conflicts with local system	Unable to install server applications

Yet the basic idea behind *Flatpak* is attractive. Find a program, install it easily, and run it without worrying about what else on your system may be using an incompatible version of the same library. Furthermore, installing applications as Flatpaks enables you to mix and match programs in ways you can't with your normal package system, such as:



A flatpakref file contains all the information necessary to find an application, determine its dependencies, and install everything.

- Using the bleeding-edge version of an application, while keeping the standard version as well. Example: *LibreOffice*, shown in the screenshot (*on page 74*) running versions 5 and 6 at the same time.
- You have security concerns, and feel better with the program operating in a sandbox. Example: Adobe's *Flash Player*.
- To use a more recent, more stable, and/or more full-featured version of an application than your distribution provides. Example: *Calibre*, which is well behind even on Ubuntu 16.04.

Yet as we saw in **LXF234**, there are other formats that provide similar advantages. These include Ubuntu's Snap format (**www.snapcraft.io)** and AppImage (**https://appimage.org**). While they all perform similar functions, *Flatpak* has a leg up on the competition in a couple of ways:

- *Snapcraft* is a Canonical project, and while it's available in other distributions, it's a first-class citizen of Ubuntu. If you use another distro, or just prefer a software system that's not controlled by a single, commercial entity, *Flatpak* is a better choice than Snap.
- *AppImage* takes the approach of simply downloading and running a single file. This is straightforward, but not necessarily easy, especially for new users. First off, you need to set the downloaded AppImage file to be executable before you can run the app. In addition, there isn't a built-in mechanism to update your AppImages. The project offers an external tool to do so, but you need to download and configure that yourself as well.

Despite all this talk of pros and cons, you can use any (or all!) of the above formats on your machine at the

QUICK TIP

Flatpaks are identified by a sort of reverse URL scheme, the same kind you might use when naming code packages. The top-level domain comes first, then the domain, then sub-domain, and so on.

same time. That defies logic, doesn't it? How did the developers accomplish this? Let's examine the Flatpak format a little more closely to find out.

Unpacking Flatpak

Like most open source projects, *Flatpak* stands on the shoulders of giants to accomplish its goals. Two key technologies behind *Flatpak* are ostree and bubblewrap.

When you install a Flatpak, you're installing an ostree "bundle" containing the application. The project's website describes it as something like "git for operating system binaries." In addition to other benefits, this approach provides for atomic updates, which is a fancy way of saying that only what needs to get updated, gets updated. It also provides a rollback mechanism.

Suppose you have a program with file 1 and file 2. An update provides a new version of file 2, which is actually designated file 3. The program will operate quite happily using file 1 and file 3 until you want to roll back a version. But you won't need to do a whole uninstall/reinstall operation. You just need to change the application so it uses file 2 instead of the newer file 3 (which still exists, in case you want to "roll-forward").

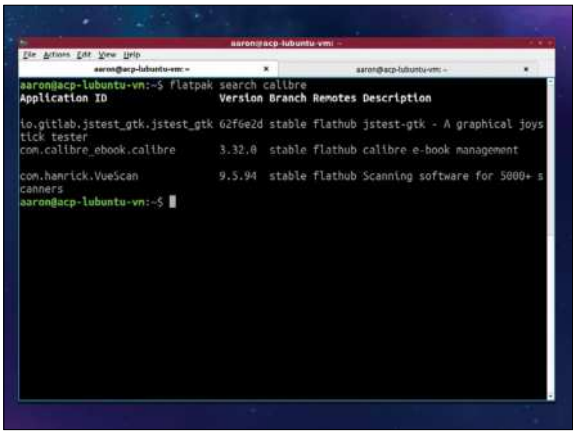
The other helper technology is bubblewrap, a software layer that in turn uses the Linux kernel's user namespaces feature. It enables non-root user accounts to serve as root accounts in containers, including a *Flatpak* sandbox. When you run a *Flatpak* application it'll start a sub-system consisting of the *Flatpak* application and a select number of system resources for the sandbox to use. The *FlatPak* wiki on GitHub (**https://github.com/flatpak/flatpak/wiki/Sandbox**) has a page that describes the contents of the sandbox in detail, but some key attributes are:

- The application's runtime(s) are mounted in the sandbox within **/usr** (the structure of which is explained in more detail in the box (*right*).
- The application itself is mounted within **/app**.
- Configurations from the host **/etc** directory are mounted to the sandbox's **/usr/etc**.
- Only processes in the sandbox are visible from the sandbox, for example the **/proc** directory.
- Assets such as icons, fonts and so on are available through "bind-mounts," or mounting one file/directory (outside the sandbox) to a different spot in the filesystem (within the sandbox).
- Overall, the sandbox has access only to the resources it needs to function, and then only with the least possible privileges.

Now, if you're not too confused yet, we'll add another level of complexity. Because while most software requires root privileges to install on a Linux system, normal users can install Flatpaks. In the next section we'll examine the difference between these "user installs" and a more traditional "system install."

System versus User

Seasoned Linux users are accustomed to providing a root password when installing software. That software ends up in a directory outside your **$HOME**, such as **/usr/bin**, **/opt** or perhaps even **/usr/local**. But as mentioned in our pros list, *Flatpak* supports the ability to install a program for a specific user, as opposed to the entire system. This works because a normal user has access to the necessary resources to run a program,



The results of the flatpak search command give you the three-part, reverse-domain ID that you'll need to install.

» ANATOMY OF A FLATPAK

Installing a Flatpak will create a file structure for the program and its runtimes, as well as some metadata. The basic structure of this consists of four top-level directories in either **$HOME/.local/share/flatpak** or **/var/lib/flatpak**, depending on whether it was a system or user install.

The first of these is **repo**, which contains information about installed applications, remotes, and other metadata. The **app** directory is where the applications themselves live. Each program has a sub-directory named with its three-part identifier (for example, **org.gnome.GEdit**). There are further sub-directories that are set aside for architecture (such as x86_64), version numbers, and their associated checksums.

Runtimes are managed in the same way, but are kept separately in the **runtime** directory. This is because they can be shared across more than one application. The final directory is **exports**, containing items related to the Flatpak programs yet sits outside the sandbox. These include things like the application icon and **.desktop** file (which will appear in your desktop's application menu), and system references such as connections to the windowing system. The **exports** directory contains links to things inside the Flatpak, and is one of the carefully controlled touchpoints between the host system and Flatpak's sandbox.

The following shows the export of the **tree** command run following a *LibreOffice* Flatpak installation:

```
$ aaron@acp-demon:/var/lib/flatpak$ tree -L 2
app
  org.libreoffice.LibreOffice
  appstream
    flathub
  exports
    bin
    share
  repo
    config
    extensions
    flathub.trustedkeys.gpg
    objects
    refs
    state
    tmp
  runtime
    org.freedesktop.Platform
    org.freedesktop.Platform.ffmpeg
    org.freedesktop.Platform.Locale
    org.libreoffice.LibreOffice.Locale
```

»

and the sandboxes created by the user receive this same access.

When you run any of the install commands we'll see in a few moments, you'll be prompted for your password. This is because *Flatpak* will carry out system installs by default. What this means is that a program will be available to all users and that its files will be stored in **/var/lib/flatpak**.

Alternately, if you use the `--user` flag with a command – it'll just install for your account (you shouldn't be prompted for a password). The files in this case will be in **$HOME/.local/share/flatpak**. This is useful if you don't have an administrator account on your machine, or perhaps if you want to try a bleeding-edge version of a program that's installed elsewhere on the system.

The user install is possible because of how the sandbox is constructed. Think of it this way: as a user, you have the ability to run an application. That application has the rights (through your user account) to create an on-screen window, write files to disk, and access network connections. When you install a Flatpak as a user, any resources from the host system are limited by what you yourself could access with a "normal" program.

Installing Flatpak

The first component you need to install Flatpaks (the package format) is *Flatpak* (the application). Most distros provide it in their repositories. In Ubuntu, you can issue the following to get up and running:

```
$ sudo apt install flatpak
```

This will, of course, give you the version that's packaged with the current version of your distro. If you're on a long-term support version, such as Ubuntu's 16.04 (Xenial), this may mean you won't get newer features. Installing from the official *Flatpak* PPA is the recommended way to make sure you always have the most recent version:

```
$ sudo add-apt-repository ppa:alexlarsson/flatpak
$ sudo apt update
$ sudo apt install flatpak
```

Once it's installed, you'll be able to install Flatpak programs using the command line. But some GUI package managers also have support for the format. KDE's *Discover*, for example, can manage your Flatpak files if you install the appropriate backend.

With the right tools installed, there's one quick step to take before you can start hunting for some new apps.

QUICK TIP

If you've just installed Flatpak and your first program, you may notice there's no icon for it in the system menu. The solution is to log out, then log back in again. This will reload the $XDG_DATA_DIRS environment variable, to which Flatpak has added its own paths.

Flatpak hubs

Flatpak seems to have adopted the term hubs to describe collections of software, just like RPM or DEB repositories. The *Flatpak* site itself hosts the first one you should add, FlatHub. Register it with your system using your package manager, or at the terminal with the following command:

```
$ flatpak remote-add --if-not-exists FlatHub https://flathub.org/repo/flathub.flatpakrepo
```

You can see a list of your configured hubs by running

```
$ flatpak remotes
```

On the menu

There are a couple of different application types that lend themselves to being packaged as Flatpaks. The first of these is commercial applications. If the developers of commercial products don't make it easy for the community to pick their applications apart, it's solely up to them to package everything.

Flatpak gives a single target for packages that will work across distros, and sandboxing the apps can help reduce support costs. Some examples of commercial (though free) programs in Flathub include *Steam*, *Slack*, and *Sublime Text*.

Another type of application that makes a good Flatpak is large, complex applications. These benefit from *Flatpak's* atomic upgrades, making the update process a little less painful, and the roll-back process even less so. *LibreOffice*, which is normally a 200MB update, and *Android Studio*, which is even more, are both available as somewhat more convenient Flatpaks.

Finally, cross-platform apps already have two completely different OS targets (at least for desktop, even more if they're mobile). Developers and packagers surely don't need to worry about the idiosyncrasies of a dozen Linux distros on top of that. Just wrap your app up as a Flatpak, and you're good to go.

Installing Flatpak programs

Installing applications from the Flatpak will feel very familiar if you're accustomed to using *apt* or either *yum*

Two different versions of LibreOffice running at the same time, on the same machine. Is it magic? No, just Flatpak.

» ROUNDING OUT FLATPAKS

This article comes at a time when Flatpak recently enjoyed two significant developments. The first is the project's v1.0 release, which came out on 20 August, 2018. As befits such a milestone, the press release (**https://flatpak.org/press/2018-08-20-flatpak-1.0**) called the new version "feature-complete," with all the necessary resources for both developers and users. So in addition to the *Flatpak* commands we've mention in this article and support by most major GUI package managers, there are also tools to help developers package their applications as Flatpaks. If you're a coder-type, check the "Build applications" section of *Flatpak's* help (**flatpak --help**) for an idea of some of the utilities at your disposal.

But the launch of another initiative is arguably more interesting. The WinePak project (**www.winepak.org**) project utilises *Flatpaks* to help ease the installation of Windows programs. It does this by providing the application and an optimised *Wine* installation in a format that's simple to install. It promises to do away with all the configuration-fiddling that typically comes along with *Wine*-based installs. While the selection at the moment is quite small, there are some top-tier games already in there, including *Overwatch*, *World of Warcraft* and *League of Legends*. Sure, by installing these you might end up with *Wine* on your system a dozen times. But they're sandboxed and storage is cheap, so go nuts!

or *dnf* on your system. The search command will show you hits in any of your configured for the keyword that you provide:

```
$ flatpak search calibre
```

This will return a result for the excellent *Calibre* e-book manager, which is often out of date in the Ubuntu repositories. Note the first column, which gives the Application ID in a sort of reverse-domain format (com first, then domain, and so on.). You'll need to give this to the install command for *Flatpak* to install your program:

```
$ flatpak install com.calibre_ebook.calibre
```

At this point *Flatpak* will start downloading and installing your application and any runtimes (org.freedesktop – see the screenshot, *below right*) it requires. A runtime is a collection of libraries and other software for the application. It's the *Flatpak* way of resolving your dependencies for you.

"Wait," you might say, "isn't this what *apt* and *yum* do for me?" It's a fair question. There are a few notable factors that set runtimes apart from normal Linux dependency management.

- *Flatpak's* sandbox environment ensures anything you install won't conflict with your system proper.
- A runtime is a fixed set of software, so you know you'll get exactly you need to run the app – no more, no less.
- Flatpaks are built against specific runtimes, so there's no need to worry about upgrades breaking your program all of a sudden.

The trade-off boils down to confidence that the program will always have what it needs to run as intended, but note that it'll need to take up more disk space to do so.

There are a couple of other methods you can use to install applications. If you happen across a rogue Flatpak and download it, install it from the command line using the command above:

```
$ flatpak install somefile.flatpak
```

Finally, *Flatpak* provides a method for installing an application by running an installation file. This file, called a **flatpakref**, contains some metadata about the program, including:

- Its full name.
- The hub from which it should be downloaded.
- Its version
- Whether it is a runtime or not.
- Its GPG signature.

Installing from a **flatpakref** file is the same as installing from a locally downloaded Flatpak file:

```
$ flatpak install ./somereffile.flatpakref
```

You can also use a GUI package manager, which should display and install Flatpaks in the same way it does other packages (provided that you installed the backend).

Running Flatpak programs

Once your program is installed, you can launch it from the command line as follows:

```
$ flatpak run com.calibre_ebook.calibre
```

Once the program starts up, there are a couple of things you'll notice. The first is that you'll have access to all your normal **$HOME** directory files. "But, but," you'll ask, "I thought the *Flatpak* sandbox was isolated from the rest of the system?" That's very astute of you. However, running a program that can't interact with your files isn't very useful. The *Flatpak* developers have worked around this.

You should also see entries for the application show up in your desktop's menu. If you don't see them, check the Quick Tip (*left*) for how to try and fix this. Actually, under most circumstances, you shouldn't be able to tell the difference between these programs and those provided by your distribution. You can operate on your files, play multimedia files, and generally use them as a "normal" program.

Managing your Flatpak installs

Once you have a Flatpak or two installed, you can, of course, use your GUI package manager to administer them. But if you prefer the terminal, some of the below commands will be useful. You can list the Flatpaks (both apps and runtimes) you have installed with the following command:

```
$ flatpak list
```

The result displays not only the installed modules, but also their installation type (**system/user**). You can update all your Flatpaks with a single command, just like native package managers:

```
$ flatpak update
```

In the event you need to remove a Flatpak, enter the following at the prompt:

```
$ flatpak remove [full-qualified app name]
```

If you need to remove one of the hubs you've configured, you can do so with the following:

```
$ flatpak remove-remote [name of remote]
```

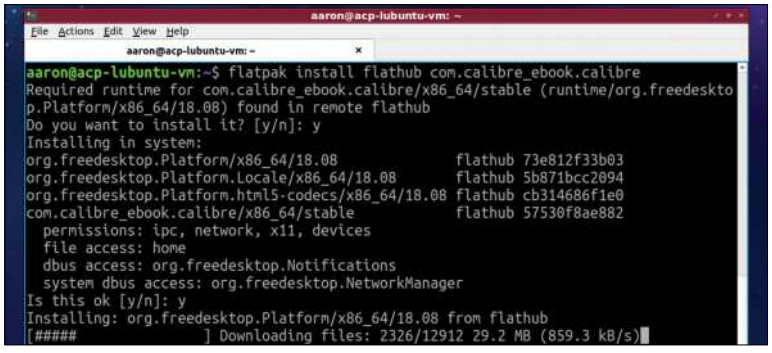
Conclusion

The benefits of the Flatpak format are numerous. It provides developers with an option to target multiple distributions, while giving users multiple tools to install the resulting applications. It enables programs to access the resources they need, while restricting them from the things they don't. It works around some of the dependency problems of traditional packaging formats, at the expense of a little disk space. Finally, it opens up opportunities for developers to wrap up applications that are traditionally tedious to install in a format that's drop-dead simple.

So if you want to try out the cutting-edge version of *LibreOffice* or run a *Wine*-based game like *World of Warcraft*, give *Flatpak* a try. You really can't go wrong! LXF

QUICK TIP

If you run the `ls /` command when a Flatpak is running and don't see any difference, run the following: `$ flatpak run --command=sh org.libreoffice.LibreOffice` **This will start a shell within your app's sandbox. Now if you run `ls /`, you should see directories such as `/app`.**



Like traditional package managers, Flatpak will download and install Calibre and the runtimes on which it depends.

Roundup

Calligra Suite » Google Docs » LibreOffice » OnlyOffice » WPS Office

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

Office suites

First desktop suites and individual packages, and now office suites have also found their way online. **Shashank Sharma** discusses the best on offer.

HOW WE TESTED...

We're running the latest version of all the applications on top of the Budgie desktop. *WPS Office* and *OnlyOffice* both provide Snap packages, so installation is a breeze. You'll also find the latest *LibreOffice* and *Calligra* in the software repositories of many distros. This is why we haven't tested them on the ease of installation. However, if your distribution doesn't feature the latest Calligra in the repositories, we'll caution you against attempting to install it manually. The compilation takes far too long, only to result in insurmountable errors.

While we've tested the three common components of an office suite, we're just as interested in other components on offer, if any. Another important factor is the support for different formats, including ODF, and even *Microsoft Office's* proprietary format, and how they render files.

Extended functionality through plugins, and the available documentation to help new users acclimatise to the different tools is also an important criteria.

Since last we looked at office suites way back in **LXF167**, the quintessential productivity applications on offer for Linux users have undergone a sea change. For one, the applications now boast of a much better compatibility with proprietary formats. Another major development is the introduction of several online office suites similar to Google Docs. In fact, many major players, such as *Microsoft Office* and even *LibreOffice*, now offer online solutions.

A desktop office suite typically comprises many different applications. There's usually an program each for word processing, working with spreadsheets and making presentations. Some suites also feature a database or a note-taking application.

Most Linux distros, even lightweight variants such as Damn Small Linux, ship with an office suite, or programs to provide the same tools. You don't always have to resort to a mammoth project if all you're interested in is a word processor. In such situations, you can also opt to install standalone applications such as *Abiword* for word processing and *Gnumeric* for spreadsheets.

What makes these applications stand out from one another is the default feature set. This becomes all too apparent when you compare an online solution with an offline desktop application.

Word processor

Let your inner Shakespeare shine!

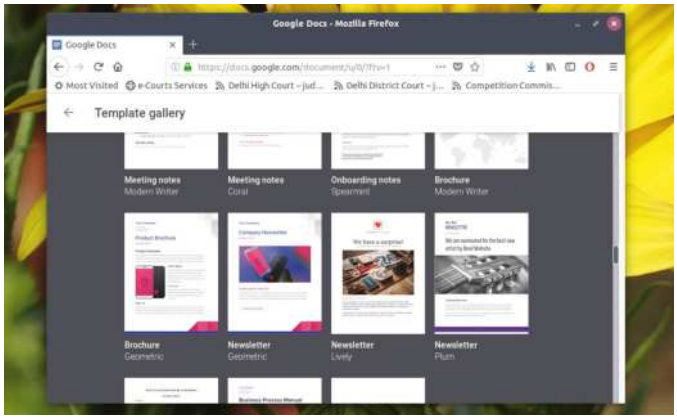
The word processor in *Calligra Suite* is called *Words*. Unlike the other word processors however, *Calligra Words* features a sidebar on the right that's used to tweak the different formatting options on the document, such as inserting comments, or headers and footers, and so on.

Apart from this default sidebar that you can't dismiss, the application also provides many different elements, called dockers. These include document statistics, add shape and more. Each of these elements takes quite a lot of the space on the screen, leaving precious little for your actual document. Despite excessive and constant resizing of the different dockers, and closing the ones you don't need, you'll be left with almost no room to even access the comments field! Because of its vastly different design, the tool isn't the easiest to work with, especially if you've ever worked with a word processor before.

Google Docs enables you to search for topics online, and drag the results into a document, automatically adding citations, but that's the extent of its USP, aside from being online.

Both *LibreOffice Writer* and *Calligra Words* boast of out-of-the-box support for various formats, including Open Document Format and *Microsoft Office's* proprietary formats. Unfortunately, both these applications tend to distort the document's appearance somewhat when working with doc or docx files. Whereas *LibreOffice Writer* miserably fails, more often than not, to identify text formatting and renders the document quite differently to the original, *Calligra Words* fares far better. Yet even *Calligra Words* sometimes has difficulty with rendering numbered lists, and messes up the margins.

Of all the applications featured in this *Roundup*, *OnlyOffice* provides the best support for *Microsoft Office* formats, and



All the projects in this month's Roundup feature a number of templates to choose from for all the component applications that make up an office suite.

flawlessly opened all the files that we tested. Unlike the other tools, the comments added to a document are displayed on the left, but the interface is otherwise quite similar to the ribbon style on *Microsoft Office*.

The biggest drawback of *WPS Writer* is that it doesn't support Open Document Format (ODF). The FAQ itself explains that the developers might consider adding ODF functionality when they have a larger team of developers, or gather enough users to justify devoting the time and effort that's required for implementing ODF support. But for now, ODF is in *WPS Writer's* pending tray.

VERDICT

CALLIGRA SUITE	6/10	ONLYOFFICE	10/10
GOOGLE DOCS	7/10	WPS OFFICE	5/10
LIBREOFFICE	10/10		

Calligra fares poorly because of its cluttered interface, while WPS Office trails in fifth place because it doesn't support ODF.

Spreadsheet

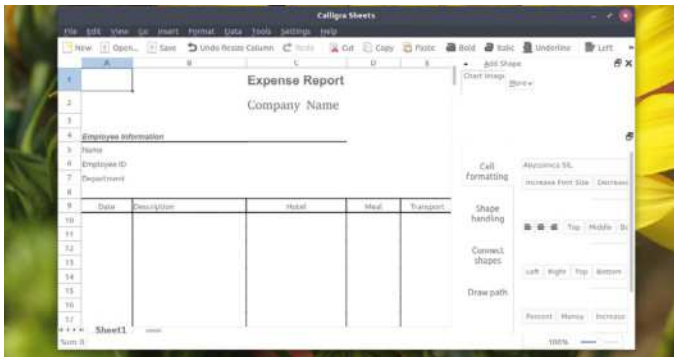
Number crunching made easy.

The spreadsheet applications on the different office suites offer nearly the same functionality. At the minimum, they each provide a number of functions to create formulas and perform complex calculations. Besides performing basic mathematical functions, all the spreadsheet applications boast of built-in functions for common financial and statistical operations. One of the best features of these modern replacements for pen and paper worksheets is what's known as what-if analysis, where you change one field of data and see the tools automatically change the values of associated fields.

LibreOffice Calc and *WPS Spreadsheets* hold the distinction of being among the easiest to use. In addition, *Calc* boasts of many wizards to help you use its advanced functions. You can easily pull data off databases, and collaborate with other users, which makes it ideal for most office setups.

Google Spreadsheets isn't that far behind, and as with all the other *Google Docs* applications, it too provides many different templates to choose from. The documentation is on point, and precise in the help it offers.

All the programs support a variety of formats and enable you to export the data as either a PDF or CSV file, and with the exception of *Calc* and *Sheets* all default to the xlsx format. The



WPS Spreadsheets offers a more populated interface than OnlyOffice, but Calligra's is the most difficult to work with, out of the products on test.

applications also look fairly alike, with *Calligra Sheets* being the only exception. Once again, the reliance on dockers to provide different functionality makes working with *Calligra* an unintuitive experience, despite the fact that it provides the same functionality as other applications, such as *LibreOffice Calc*.

VERDICT

CALLIGRA SUITE	7/10	ONLYOFFICE	9/10
GOOGLE DOCS	10/10	WPS OFFICE	10/10
LIBREOFFICE	10/10		

Once again, Calligra suffers because of its interface and not the features on offer.

User experience

They all look alike, but do they all behave the same too?

Productivity applications is a term that gets thrown around quite a bit nowadays. Until the 1980s, the world was still getting by on typewriters or pen and paper, but with the introduction of office suites, the oldest productivity applications changed how we use the computer at home and work. It's no surprise, then, that there are very few jobs in the world today that don't require basic skills with office software.

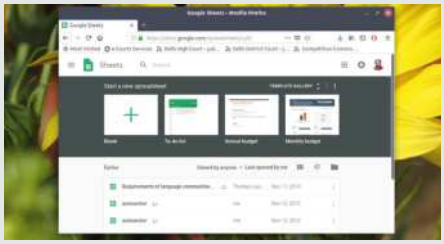
While the suites all provide similar functionality in one or two component applications, they are distinct enough in their workflow that switching between them may involve quite a bit of learning before you're comfortable using them for everyday use.

If you're only interested in a single office component, such as a spreadsheet or word processor, then refer to the Also Consider section at the end of this month's *Roundup* for some recommendations.

Google Docs 8/10

It has the distinction of being the only online office suite featured in this *Roundup*. But to be fair, *OnlyOffice* too provides a cloud-based solution. While originally the suite worked best on the *Chrome* browser, you won't find any performance issues or missing features when using it on alternative such as *Firefox*.

The vast number of templates for each of the component applications is the biggest USP for *Google Docs*. Its easy integration with Google to perform searches, and integrating results, or sharing files with other users with a single click, is also a bonus. If you frequently have to collaborate with users, or are regularly on the move and can't afford to leave urgent files behind, the use of Google Drive as the storage provides great luxury, in that you can access your work from anywhere – as long as there's an available internet connection.



Calligra Suite 7/10

There's little wrong with *Calligra* itself, but you may struggle with its esoteric workflow. For instance, you can't insert any shape into a presentation or document without accessing the Add Shapes docker. Indeed, the whole process of switching between dockers to access elements and features that are easily accessible on the alternate suites from the toolbar itself, is inexplicable and unintuitive.

We don't generally condemn the different style of working, but easy availability of documentation becomes quite important for such applications. While plenty of documentation exists for some of its component applications such as Sheets and Stage, it's not accessible from within the app itself.

But if you're willing to put in the time to master its style of working, or are using old machines, you'll appreciate Calligra's ability to function on low-spec systems.

Help and support

Who can you turn to, when you can't tell if you're coming or going?

Judging by the nature of work that these programs perform, you'd think they're fairly easy to use. While this is true for the most part, especially for the straightforward offerings of *Google Docs*, the other applications may require some guidance. Such as when you're trying to figure out how to install plugins on *WPS Writer*, or if there are even any on offer!

While all the applications are cross-platform, *WPS Writer's* Linux edition doesn't get the same attention as its Microsoft counterpart. This is why you may find solutions to certain problems in the official documentation that don't apply to the Linux edition. For instance, the official Help will tell you how to use the Mail Merge feature, but it isn't even available on Linux.

There isn't much documentation on *OnlyOffice's* website, beyond instructions covering installation and basic usage, but you'll find far more useful information from within the program itself. Depending on the application you're running, clicking File>Help presents a categorised list, and there's also a search bar if you're looking for something specific.

In contrast, *Calligra's* documentation is non-existent. Launching the handbook from within the program drops you to a KDE handbook page that doesn't exist. When you do finally manage to track down some documentation on KDE Userbase, all it lists is a few isolated features. Thankfully, you can tap the user community for advice and assistance on the official forum boards.

LibreOffice boasts of detailed user guides, accessible from within each of the component applications. In addition to its thorough documentation guides, the project also has a Q&A website called Ask LibreOffice.

VERDICT

CALLIGRA SUITE	4/10	ONLYOFFICE	8/10
GOOGLE DOCS	8/10	WPS OFFICE	4/10
LIBREOFFICE	10/10		

Office suites are unlikely to attract new users if there aren't any help files to help acclimatise them.

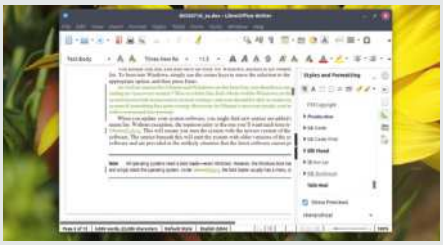
LibreOffice 10/10

One of the most comprehensive suites, each of the component applications feature detailed documentation that can be accessed from within the app itself.

With the 6.0 release, the project has also introduced a ribbon interface, much like *Microsoft Office* but it isn't enabled by default. If you like the ribbon interface though, the implementation on *WPS Office* is quite pleasing, but unfortunately not so on *LibreOffice*.

Despite its ease of use, our only complaint with *LibreOffice* is how poorly it renders most docx files. Another frequent complaint is that it's a resource hog, and if you're running a low-spec machine with only about 2GB RAM, you'll probably be best served by one of its alternatives.

But it makes up for these shortcomings by offering hundreds of extensions which enhance its functionality, and *LibreOffice* is unmatched on this front.



OnlyOffice 8/10

For this *Roundup*, we've tested the *OnlyOffice Desktop Editors*, which is the offline variant of the office suite. This version doesn't provide many useful features such as collaboration. You can try various cloud installations with a free 30-day trial, if you want to experience the whole host of features. While these variants might suit enterprises, or even a SoHo setup, they're much too expensive and cumbersome for home users.

Unlike the other suites that feature separate applications, *OnlyOffice* presents a single interface. You can decide whether you wish to work on a presentation, spreadsheet or document, or all three, at any time, which open up in separate tabs within the interface.

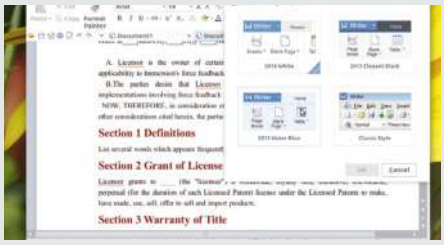
For basic home users however, *Only Office* provides most of the common functionality you expect from three of the most common productivity applications.

WPS Office 7/10

Both *Only Office* and *WPS Office* only offer a spreadsheet, word processor and presentation programs, while the others also provide a drawing application, or a database management tool. Of the two, we like *OnlyOffice* better because the project is more mature in comparison, since *WPS Office* is lucky to be able to push out a Linux release.

With a barebones team working on the Linux edition, we're also unsure if the project will still be available down the road. It's also unknown, when if ever, it will introduce some of its best features, such as collaboration, which are restricted for the Windows version for now.

The lack of support for ODF is also disappointing, and might put off many open source purists. But its ribbon implementation is impressive, and we'd go so far as to suggest that this is sometime *LibreOffice* can learn for *WPS Office*.



Other tools

What else can you accomplish with these programs?

Word processor, spreadsheet and presentation application are three of the most commonly used tools in an office suite. Our selection of suites for this *Roundup* each offers other applications as well.

Google Docs's Forms can be used to quickly create some of the most common ones, such as contact, RSVP and party invite. The templates provide a vast selection of the commonly used ones across many different categories such as Personal, Word (job application, event feedback), and Education (assessment, worksheet). You can also use *Google Drawings* to create flowcharts and mind maps, but not artistic works.

Calligra provides you with *Karbon,* a vector graphics drawing application, but the tool is no longer maintained. If you're interested in a graphical application for building databases, you can try your hands at *Kexi*. *Calligra Plan* is a project management tool that can be used to manage even moderately large projects.

Like *Calligra's Kexi*, *LibreOffice Base* is a graphical database management application similar to *Microsoft Access*. *Base* boasts of various wizards to help new users, or even skilled hands create tables, queries, forms and reports. It also features pre-defined table definitions for tracking assets, customers, sales orders and invoices to ease quick deployment. It provides native support for most of the widely used database engines, such as MySQL, MS Access, PostgreSQL and others. In comparison, *Calligra's Kexi* doesn't support *MS Access* and isn't as well developed as *Base*.

Unlike the other tools, *WPS Office* and *OnlyOffice* only feature the three essential components, and there's no word on whether they'll ever provide a drawing or database application.

VERDICT

CALLIGRA SUITE	9/10	ONLYOFFICE	5/10
GOOGLE DOCS	7/10	WPS OFFICE	5/10
LIBREOFFICE	9/10		

The scores here only reflects the number of tools on offer within each suite, and not their usability.

Collaboration

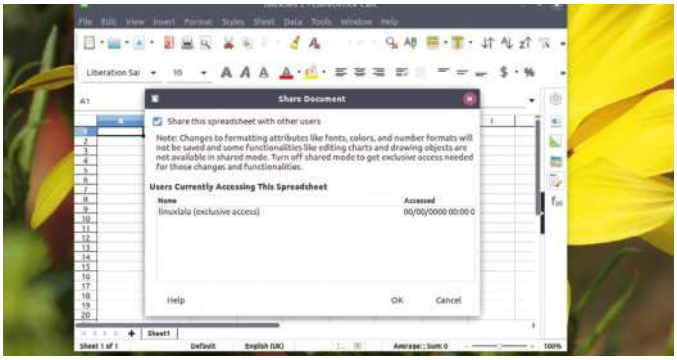
What are your options when you need to share the workload?

A major factor for the popularity of online office suites is that they enable multiple users to collaborate and work on the same document simultaneously. While offline office suites are catching up, this is one domain that's ruled by online office suites, such as *Google Docs*.

Google Docs enables real-time, character-by-character collaboration on *Google Docs*, *Google Sheets*, *Google Slides* and *Google Drawings* tools. When multiple people are working on the same document as the same time, they'll all be able to see the changes that have been made by each other. These collaboration options work in conjunction with the sharing options that makes it possible to set access levels for files, so that you can control who sees and edit your files. Again like with other features, you can use another Google service, Google Groups, to share the document with a group of people with a single click.

LibreOffice Calc enables document sharing with simultaneous write access for many users. Assuming each user adds a name on the *LibreOffice* User Data page under Tools>Options, it's quite easy to track the contributions made by the different users.

A similar feature has still not been implemented by the *Calligra Suite* and there isn't any mention of it being introduced in future releases. The project's website doesn't have a roadmap for future releases, so there's no telling when, if ever, this feature might be introduced.



Collaboration on LibreOffice is restricted to Calc, as the other apps don't provide simultaneous write access.

With *OnlyOffice*, you get a visual cue informing you of the passages your collaborators are working on, and you can talk to them to discuss ideas. But this feature isn't available on the offline editor, and you must use the cloud feature to access it.

VERDICT

CALLIGRA SUITE	N/A	ONLYOFFICE	N/A
GOOGLE DOCS	10/10	WPS OFFICE	N/A
LIBREOFFICE	8/10		

Collaboration is yet another feature that's yet to be implemented on the Linux version of WPS Office.

Presentation

A good slide-show takes more than just a collection of bullet points.

As with its word processor applications, *Calligra's* presentation program *Stage* also takes some getting used to. The different formatting options, such as defining the animation and transitions are relegated to a separate docker, on the right of the screen. Adding shapes is similarly confined to the Add Shapes docker.

Google Slides offers the choice of a number of templates to help you get started. You can add pictures, shapes and charts to the slides, and configure transition effects for each. It's also possible to choose different colour schemes and themes to create dynamic slides. Also on offer are a large number of add-ons that mean you can add even more content to a slide. For instance, you can insert free HD photos to your slides using the Unslpash add-on. The add-ons are split into different categories such as Education, Business Tools and Productivity.

OnlyOffice enables you to choose from among various background designs. You can introduce shapes, charts, text art and pictures, and add transitions to the slides. A unique feature is the built-in photo editor. It can be used to manipulate the brightness, contrast and sharpness of any image inserted into a slide. You can also add a frame to the image, crop, resize, change the orientation and do many other edits. The photo editor is slower, in comparison to dedicated image-editing tools like *Gimp*. But it gets the job done, and the convenience of editing images from within the tool itself cannot be overstated. You can also insert a YouTube video into a slide by clicking the YouTube tab under Plugins and providing the URL of the video. However, the video doesn't run when you start the slideshow and all you see is an image of the video, with no button or any option to start playback.

The rehearse timing feature on WPS Presentation lets you time your presentation for automatic transition for each slide.

WPS Presentation and *LibreOffice* offer all the functionality you expect from a presentation tool, such as adding animations, transitions, designs and so on. Both *WPS Presentation* and Calligra Stage enable you to choose the slide size (either standard screen 4:3 or wide screen 16:9).

VERDICT

CALLIGRA SUITE	5/10	ONLYOFFICE	10/10
GOOGLE DOCS	9/10	WPS OFFICE	10/10
LIBREOFFICE	10/10		

The biggest drawback of Calligra is its unconventional interface.

Office suites
The Verdict

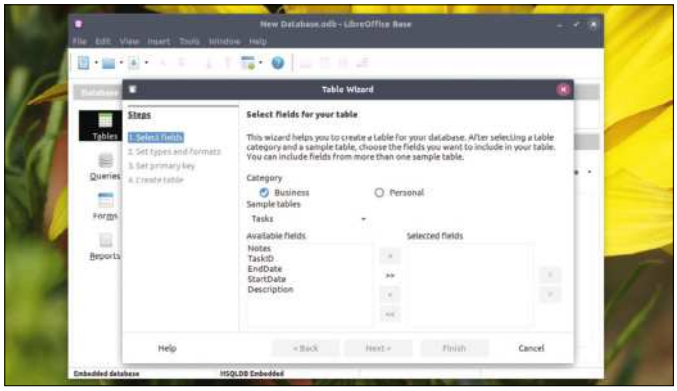
The choice of *Roundup* subjects over the past several issues has left us to conclude that perhaps one application might not cater to all your needs. This was true for the image and video editors, and office suites aren't an exception either. If you can spare the resources, you might find yourself preparing presentations on one suite's application, while relying on another for your spreadsheets.

With the exception of the podium finishers, the other office suites alternately disappointed and impressed us in equal measure. *WPS Office* is a highly capable suite, and we would optimistically advice users to give it a chance, but the lack of ODF support, with no word on when if ever it will be available, sealed its fate for this *Roundup*. Never mind that its fast and responsive, and fun to work with. The lack of many useful features on the Linux edition also goes against it.

Calligra Office, despite being highly capable, and offering an impressive feature set, was also a major disappointment – but chiefly for its interface. If you can get past its esoteric workflow, and don't mind finding your own way through trial and error, there's nothing per se wrong with it. But if you plan on running *Calligra* on a small screen, say a 14-inch laptop, you'll barely have any room to work on your document or slides, and so it too couldn't make the podium.

While both *OnlyOffice* and *LibreOffice* support the proprietary *Microsoft Office* formats, *OnlyOffice* is far better at rendering documents. Unlike the other suites, however, it doesn't feature a database or drawing application, only offering a spreadsheet, word processor and presentation application. The offline desktop edition also doesn't provide all the same features as are on offer with the cloud variants, and that's disappointing too. Still, for the limited number of applications on offer, *OnlyOffice* performs rather well, and isn't as resource hungry as *LibreOffice*. If you don't care for collaboration then we would definitely advise you to give it a try.

There's very little separating *Google Docs* and *LibreOffice*, save for the fact that the former needs an internet connection to function, while the latter is a desktop solution. LXF



1st LibreOffice 10 — 10/10

Web: www.libreoffice.org Licence: MPL 2.0, other open source
Version: 6.1.2
This mature office suite excels at just about everything.

2nd Google Docs — 9/10

Web: https://docs.google.com Licence: Proprietary
Version: N/A
Makes collaboration a breeze.

3rd OnlyOffice — 8/10

Web: www.onlyoffice.com Licence: AGPLv3
Version: 5.1.29
Impressive support for proprietary formats, and works as advertised.

4th Calligra Suite — 6/10

Web: www.calligra.org Licence: GPL
Version: 3.1.0
Just as feature-rich as *LibreOffice*, but its interface is very cumbersome.

5th WPS Office — 5/10

Web: http://wps-community.org Licence: 10.1.0
Version: Proprietary
Purists will dislike the proprietary licence and the lack of ODF support.

» ALSO_CONSIDER

There aren't any other open source suites to consider save for the ones already featured in this *Roundup*. If none of our solutions work for you, check out the *Apache OpenOffice* suite. *LibreOffice* is based on it, but whereas the latter has continued to innovate, *OpenOffice* has stagnated in recent years. Based on *Softmaker Office 2018*, *FreeOffice* is another freeware suite, but as with *WPS Office* and *OnlyOffice*, it only features a spreadsheet, word processor and presentation tool.

If you're looking for a cloud-based alternative, there are a couple of others besides *Google Docs*, including the proprietary *Zoho Office Suite* and the host-your-own open source *Tiki*. The AGPLv3 licensed *Feng Office Community Edition* is another viable option. If none of these catch your fancy, you can grab individual applications, such as *Gnumeric* or *Pyspread* for spreadsheets. *Abiword* and *Sozi* can similarly be used for word processing or creating presentations, respectively.

Roundup

Claws Mail » Evolution » Kmail » Mailspring » Thunderbird

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

Email clients

Can't decide which desktop email client to use? **Shashank Sharma** has been locked in the *Linux Format* Server Dungeon tweaking his SMTP.

HOW WE TESTED...

Unlike the last time we ran a *Roundup* of email clients (**LXF151** – seven years ago!) when we installed the different applications on their native desktop environments, we're running all of them on the Budgie desktop this time around. Most distributions carry these applications in their software repositories, and we're running the latest available versions of each of them. The only exception is *Thunderbird*, because its latest release isn't yet available in the software repositories, and we had to install it manually.

Most email clients are similar in appearance, but we'll be testing these tools on their performance, their search capabilities and ability to handle junk and spam messages. We'll also keep an eye out for any standout features on offer, and their support for IMAP and POP accounts. Extending functionality through plug-ins and other customisation options is another criteria to consider when choosing an application that you'll rely on for important communications.

While email clients may appear to have not changed much since their debut on the Linux desktop, the different applications are constantly working on adding new features to please existing users, and attract new ones. In addition to accommodating for touch devices, many applications also support advanced features such as delivery notifications. This is especially useful when working with an IMAP account, such as that of GMail.

Much like the perfect web browser, or text editor, there's no clear answer as to which is the best email client. The ideal feature set for a desktop email client depends on the intended use. A casual home user probably doesn't have any need for features such as delivery confirmation or scheduling, which would be relevant to business users.

Despite many viable alternatives on offer for users who prefer the command-line over a GUI, we've restricted ourselves to graphical applications for this *Roundup*. While our selection may seem heavily in favour of GTK, with only one KDE application thrown in the mix, the chosen applications can be installed on any desktop, and your distribution's software management tool will fetch the additional libraries needed for these applications to function properly.

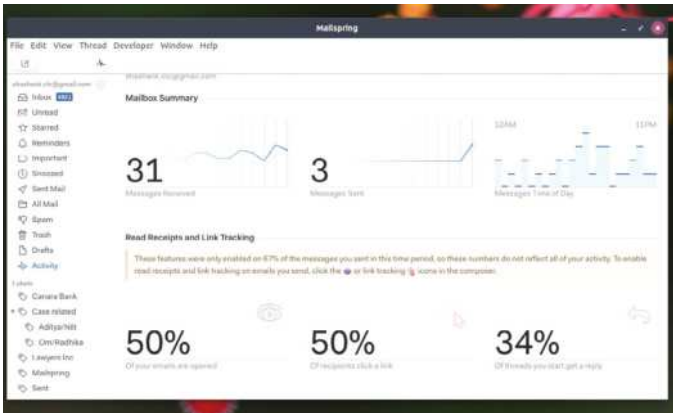
Unique features

What makes these particular email clients stand out from the crowd?

Linux machines fitted with either 8 or 16GB of RAM are unlikely to be affected by the memory footprint of the always-open email applications. However, for low-spec or older machines that have limited hardware resources, *Claws Mail* is the perfect solution. Not only does it require far fewer resources than its peers, it's also feature rich, supporting both POP and IMAP accounts. In addition to extensive filtering capabilities, it also enables you to create coloured labels when sorting messages. Setting up new accounts is also straightforward in all the clients tested here.

Apart from emails, you can also use *Evolution* to schedule appointments and track your tasks list. Together with *KMail*, both programs are especially conscious of users' privacy and provide easy means to sign, encrypt and decrypt messages. The import wizard on *KMail* supports a large number of applications that the tool can import from, such as *Evolution*, *Thunderbird* and *Outlook Express*. If you're on a non-KDE system, you must install the `akonadi-import-wizard` package to access the feature.

Like the others, *Mailspring* also supports creating multiple accounts. Even the free version provides limited access to some of its Pro features such as delivery notification, which produces a pop-up as soon as emails are delivered to the recipient's mail box. For important messages that require urgent replies, you can also create a reminder when composing messages. The application will inform you if no one has replied to your message within the specified time period. You can also receive a summary of messages sent and received, as well as track whether recipients



Many features are enabled by default on Mailspring, including signature. You should customise it before you start using the application for professional communications.

actually open your emails, and whether they click a link in the message body, by clicking Activity on the sidebar.

When reading emails with *Thunderbird*, if you double-click a message, it opens in a separate tab. Unless you close such tabs after reading the mail, *Thunderbird* will keep them open indefinitely. Unlike the other tools that require add-ons to protect you from phishing attacks, *Thunderbird* natively offers this feature. Like its brother *Firefox*, *Thunderbird* features a vast array of add-ons that can be used to extend its functionality. You can also configure it to block HTML messages, if you find them a nuisance.

VERDICT

CLAWS MAIL	7/10	MAILSPRING	10/10
EVOLUTION	7/10	THUNDERBIRD	8/10
KMAIL	7/10		

Mailspring takes the lead because of some innovative offerings out of the box.

Filter and search

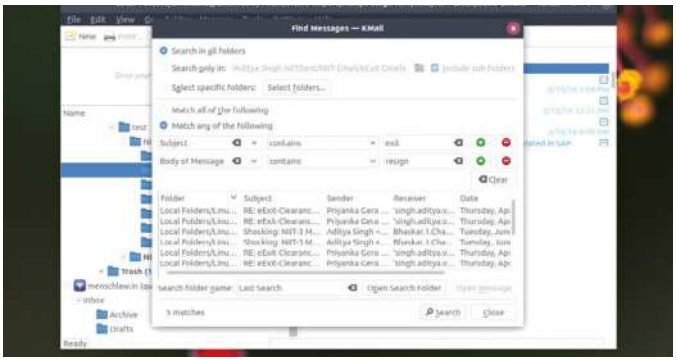
Just like Hogwart's sorting hat!

GMail now offers users' 15GB mailboxes. Yahoo similarly offers 1TB of storage space for your messages and attachments. While this is much appreciated, the downside to the vast storage space is that people no longer delete emails. Most email services now enable you to archive mails, or perform a similar function such that the Inbox is reserved only for important messages. This inevitably leads to massive mailboxes that hold tens of thousands of messages over the years.

As you subscribe to more mailing lists and add more contacts, your mail client should make it possible for you to perform filtering operations to automatically sort the different messages into dedicated folders, or apply labels for easy identification. Just as important is the search capabilities, because looking for one particular email from your father from several years ago, when he routinely sends several in a day, can be a daunting ask.

All the clients enable you to create different filters for each configured account. You can set these applications to perform a number of operations such as moving mail to specified folders and applying labels based on different conditions such as from, to, header, BCC, subject, size, attachments, and so on.

When searching for messages, *Mailspring* provides instantaneous results, offering matching suggestions even as you



Despite being on a par with all the other three in terms of speed, KMail's 'find messages' feature is disappointingly unintuitive.

type in the search box. You can also use advanced GMail-style search queries with the tool, such as 'in:unread', etc. *Claws Mail* can be quite slow when connecting to your existing mail account which already has thousands of messages. The search, although fast, lags behind *Thunderbird* and *Mailspring*.

VERDICT

CLAWS MAIL	9/10	MAILSPRING	10/10
EVOLUTION	9/10	THUNDERBIRD	9/10
KMAIL	8/10		

While all the tools are fairly fast when searching, Mailspring has the edge.

Assessing a client's usability

Will they help or hinder your work?

While online services such as Yahoo and GMail continue to tweak their interfaces every few years, desktop mail clients are far more lax with their appearances, with seemingly no change having taken place since the introduction of the modern email clients decades ago.

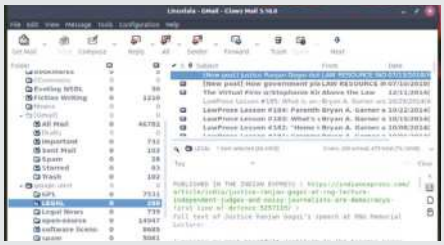
How do you then explain the popularity and continued existence of desktop clients? One of the reason for this is their tight integration into the desktop, which means pop-up notifications for incoming messages. Add to that features such as queueing messages, their anti-spam measures and support for encrypting your messages, and it's easy to see why these applications can boast of such vast user communities.

The first and only task that a mail client should do well is connect easily with the different accounts and send and receive messages. We want a client that can do this, but doesn't make the experience of putting its myriad features to use tiresome.

Claws Mail 7/10

The interface features a sidebar on the left that displays the count of read, unread and total number of messages in each folder. You can tweak this setting by clicking View>Set Displayed Columns>In Folder List. If you dislike the default layout, you can also switch to another from View>Layout, and choose from one of the five different options. There are also several other themes that you can download from the website, but these will only change the icons within the interface. Still, some of the themes can add a bit of colour to an otherwise drab appearance.

You can apply different colour labels to your messages. You can even use this colour as a search criteria when looking for messages. Instead of a globally accessible search bar, the tool expects users to select a folder in the sidebar and then click Edit>Search Folder to launch the search dialog.

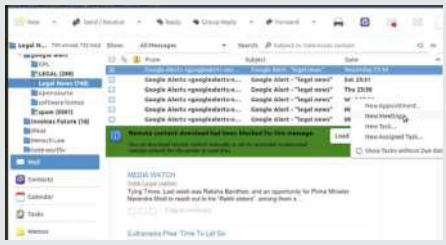


Evolution 8/10

The interface is liberally sprinkled with buttons, tabs and sidebars. The intuitively designed application is fairy easy to master, even for absolute novices who have never used a desktop mail client.

The default installation gives you access to all its features, including Calendar, Tasks and Memos. The application defaults to the mail interface, but you can switch to the other tasks using the buttons on the bottom right of the left sidebar. The sidebar at the right displays all your upcoming appointments, and you can create new ones by right-clicking a date.

By default, the application defaults to a threaded message view, where all replies to an email are collated together. You can toggle off threaded view by pressing Ctrl+T. In fact, a vast number of operations can be carried out using the default keyboard shortcuts. Click Help>Quick Reference for a complete list of these.



Configuration and documentation

Navigating the path between getting started and getting stuck, and beyond.

For this test, the awarded score is cumulative from the quality of documentation to the ease of configuring new accounts. Unlike the other tools that work flawlessly with most IMAP and POP accounts, GMail considers *Claws Mail* as an unsecure application. You're required to enable access to 'Less secure apps', from within GMail before you can configure *Claws* to fetch messages. Failure to do so results in a login failure message, even when you provide the correct password. Unfortunately, despite the user manual and FAQ on the website, the *Claws* documentation doesn't discuss this hiccup of working with GMail.

Evolution is similarly difficult to configure with Yahoo, but like the other tools, the program can automatically fetch all the pertinent details when configuring GMail. You can access the thorough documentation from within the application itself by pressing F1. The quick reference PDF provides all the keyboard shortcuts you can use.

As with most KDE applications, *KMail* has a complete handbook discussing all its features. It can connect with a large number of services without you having to manually enter any settings, and only took us a matter of seconds to connect not only with Yahoo and GMail, but also a mail server hosted on GoDaddy!

Apart from thorough documentation covering different aspects of its usage, you can also access tips and tricks and details about customisation options and more on *Thunderbird's* support page. Since *Mailspring* compulsorily requires creating an account, it regularly sends emails informing users of its myriad features. Like the others, its online help provides easy-to-follow instructions to configure, enable and use the different features.

VERDICT

CLAWS MAIL	8/10	MAILSPRING	9/10
EVOLUTION	9/10	THUNDERBIRD	10/10
KMAIL	8/10		

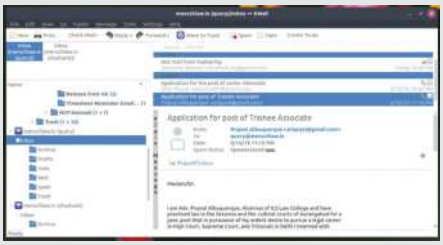
Only Evolution provides in-built documentation. The other applications launch the help section on their websites.

KMail 8/10

You can drag and drop all your regularly accessed email folders into the box on the top-left of the application's interface. This creates a list of frequently accessed folders, saving you the time to navigate the left sidebar looking for these folders.

All your emails are grouped by day/month/year. Messages are collapsed into the respective day/month/year heading, and you'll have to double-click the heading to get a list of messages. *KMail* does the threaded view better than the other tools, using line spacing to denote replies to messages in a thread. The search bar at the top can be used to perform basic search operations, but if you want more functionality, you must click Tools>Find Messages first.

For messages that require some action, click the Create To-Do button on the menu bar at the top. You can then fill in all the pertinent details to create a reminder.

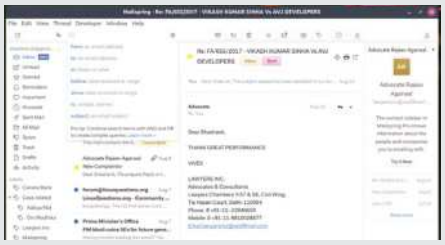


Mailspring 10/10

Unless you're comfortable with announcing to all your peers that the mail has been sent using *Mailspring*, the default signature needs immediate attention, Not quite the same as 'Sent from my iPhone', but it's still rather uninspiring.

Like *Claws* and *Thunderbird*, *Mailspring* supports theming. Click Edit>Change Theme to cycle though the six options available in the default installation.

Unlike the other tools, *Mailspring* sports a four-panel layout. The left-most sidebar is the same as on all the other applications. Next to it is the list of all the messages in the selected folder. To its right are the selected message. Finally, in the fourth pane you get details about all the people in the conversation, but it only informs you of the names of the parties in the conversation. At the bottom of this right sidebar is a list of all past emails exchanged with the same person.



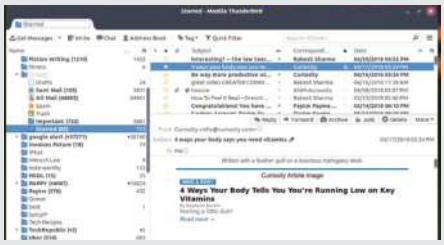
Thunderbird 10/10

Despite offering the choice of several layouts such as Wide View, Vertical View and Folder Pane, the best appearance is that of the Classic View, which presents a three-panel interface.

When confronted with an unfamiliar word in an email, you can select and right-click the word and then click 'Search Bing for "word"'. You can change the default search engine from the Preferences dialog.

When searching messages, results show up in a new tab, with filters on the left sidebar that help segregate the different messages, based on whether you were the sender or recipient and whether the messages have an attachment.

When composing a message, you can click Options and toggle Return Receipt if you want delivery confirmation. You can configure this for all outgoing messages by clicking Menu>Preferences>Preferences>Advanced. Then click Return Receipts.



Customising with add-ons

Because building on top of the stock install isn't enough...

Mailspring's default installation is the sum total of what's on offer. You're provided with a few more features if you opt to subscribe to the Pro version, at a cost of $8 per month (£6), but it still doesn't give you as many features as you can get with the other tools.

Claws is a straightforward email client, and you must resort to its plug-ins if you want additional features, such as a news aggregator. One such add-on is the ACPI notifier plug-in, which can light up the new email light on some laptop models, from Acer and Asus, for example. You can also opt from one of several themes to change the appearance of the interface.

Evolution's default installation already has several plug-ins installed and enabled. In fact, many of its basic email features, such as new mail notification is also provided through a plug-in.

A lot of functionality on *KMail* is driven with what the application itself identifies as plug-ins. Click Settings>Configure KMail and click Plug-ins on the sidebar for a list. You might be especially interested in the ad-blocker plug-in.

Thunderbird offers the choice of dozens of themes to change the appearance, in addition to specific extensions to tweak the menu entries and make other changes to the interface. You also get functionality such as the Automatic Dictionary. This extension can be used to remember the language you use when communicating with a particular person or group and automatically switches the spell-checker. The add-ons page provides a categorised list of available extensions such as Contacts, Folders and Filters, Message and News reading, Privacy and Security, and Tags. You can also scroll through the Most Popular list on the left.

VERDICT

CLAWS MAIL	7/10	MAILSPRING	N/A
EVOLUTION	6/10	THUNDERBIRD	10/10
KMAIL	6/10		

Thunderbird's vast collection of extensions puts all the others to shame.

Security features

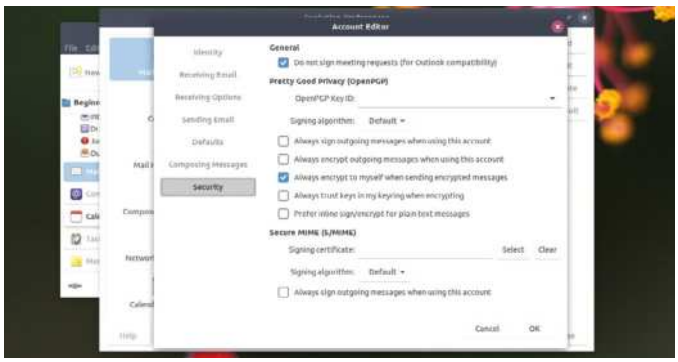
Protecting against the many hazards of the internet.

All the applications featured in this *Roundup* provide different tools and solutions to safeguard your privacy of your communications and the safety of your machines. In addition to the default settings, most also offer additional mechanisms to protect you with the aid of plug-ins.

You have the choice of using *Bogofilter* or *SpamAssassin* to detect and eliminate spam messages when working with *Evolution*, *KMail* and *Claws*. With *Claws*, the core package includes several useful plug-ins including *Bogofilter*, but if you've installed it using your distribution's software management utility, you must additionally install the *claws-mail-plugins* package.

KMail's anti-spam and anti-virus wizards help you to configure the respective tools. You must already have the tools installed to handle these operations, such as *Bogofilter* or *ClamAV*. If no tools are found, the wizard will merely inform you so. It doesn't recommend any tools. Some users have reported problems with the wizard detecting their anti-virus tools. Worse still, when connected with an anti-virus utility, *KMail's* performance is considerably slower, as each message is scanned for viruses.

Thunderbird has an advanced junk mail filter, which learns and improves its filtering depending on what you mark as spam. All incoming messages pass through the filter, and you can get



Even though it says OpenPGP, it's possible to configure Evolution to work with GPG as well.

Thunderbird to warn you about potential phishing emails, and also when a link in a message is leading you to a website other than the one indicated in the URL.

With the exception of *Mailspring*, all the clients can be configured to work with a spam detection and encryption, via the popular *GnuPG* utility.

VERDICT

CLAWS MAIL	10/10	MAILSPRING	N/A
EVOLUTION	10/10	THUNDERBIRD	10/10
KMAIL	10/10		

Despite repeated feature requests, Mailspring still doesn't have any encryption or anti-spam capabilities.

Other features

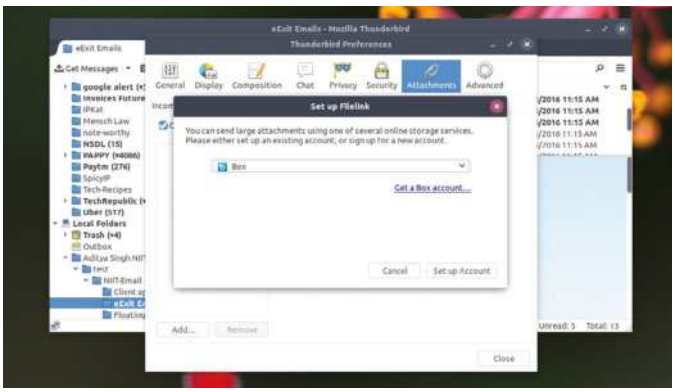
Will it make you breakfast?

Along with the web browser, the email applications are one of the most recognisable components of the desktop. What started as a means to easily communicate with your contacts have now expanded to behemoth projects with a vast feature set. Not all users would want their email client to set up reminders and double up as a calendar. But if these are things that you use daily, it's probably best to use just the one tool, rather than several.

Evolution, Gnome's official Personal Information Manager (PIM), provides a calendar, address book and a task manager. You can get the same features with *KMail*, provided you run it on top of the KDE desktop, where it can connect with other well-known KDE components such as *Kontact* and *Korganizer*.

If you frequently collaborate with non-English speakers over email, you'll appreciate *Mailspring's* ability to translate messages written in English into Spanish, Russian, simplified Chinese, French, and German. It can auto-detect your language and performs spell-check without changing the language settings. It's also especially tailored for touch devices and supports many configurable gestures such as swipe left or right.

Unlike the other tools, *Thunderbird* can also connect with chat accounts on various services so that a single application keeps record of all your communications with your contacts. The advantage of its multiple-channel chat support is that you can use the search feature to go through past conversations and emails exchanged with all your contacts. Should you ever have the need to perform a search on the web, you can do so from the search



As many services block email messages based on size, you can configure Thunderbird to connect with your account on Box.

bar within the application itself, instead of switching over to a web browser, and can even choose the search engine you wish to use.

Claws doesn't provide as many features out of the box but you can get almost all the same functionality with the use of plug-ins. It can easily double as a news aggregator, manage appointments, and more. *Evolution* and *Thunderbird* are more resource hungry than the others. Something to consider if you decide to use these applications for more than just email.

VERDICT

CLAWS MAIL	10/10	MAILSPRING	10/10
EVOLUTION	10/10	THUNDERBIRD	10/10
KMAIL	10/10		

The use of anti-spam and anti-virus plug-ins does affect performance.

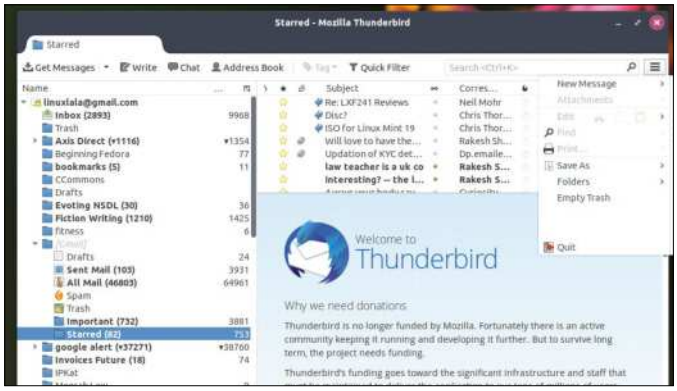
Email clients
The Verdict

Mailspring is the youngest tool featured in this *Roundup*. All the others have received extensive coverage over the past several years, and we've certainly discussed them at length over the past few pages. In a break from tradition with our *Roundups*, where we normally spend some time going over the pros and cons of the winners, this time around we'll spend more time talking about the last place finisher than all the other tools. This is because we've rarely come across an application that has so thoroughly impressed us. At the same time, a few poor choices and the lack of interest in providing key features also makes it one of the most disappointing tools ever featured in a *Roundup*.

Judging solely by the score on the different tests, you can see that *Mailspring* lags behind the other tools only in the add-ons and security features department. In today's age, an email application that doesn't enable you to encrypt messages or provide any means to filter spam is unimaginable. It's for this reason that *Mailspring* finishes last. Otherwise, we were stunned with the speedy performance of the tool. Its search feature, quick configuration and near-instantaneous launch is something you must experience yourself, as words can't do it justice. Even though the project relies on funding through the paid pro version, it still offers limited access to some of its best features such as delivery notification to the free users as well, which is much appreciated.

Granted, some of these features, such as delivery notification are also available on *Thunderbird*, but it isn't as neatly tied into the desktop was *Mailspring* on our Budgie installation, with pop-ups informing us when a mail was read, or when a person visited a link we had shared.

KMail has made stunning improvements since last we looked at it in **LXF151**, and configuring a new account is now fairly easy. *Claws Mail*, unfortunately, is still not close to the simplicity of *Thunderbird* and *Evolution*. That said, there's nothing wrong with *Claws Mail*, and if you're restricted to a low-spec machine, we would recommend nothing else. *Evolution* is a close second, but *Thunderbird* wins on account of its default features, search prowess and numerous extensions.



1st Thunderbird 10/10

Web: www.thunderbird.net
Licence: GPL and others Version: 60.0
Should look to *Mailspring* for inspiration.

2nd Evolution 9/10

Web: https://wiki.gnome.org/Apps/Evolution
Licence: GPL and others Version: 3.28
Needs to work on the account setup process.

3rd KMail 8/10

Web: http://userbase.kde.org/KMail
Licence: GPLv3 Version: 5.7.3
Could learn some tricks from *Thunderbird* and *Mailspring*.

4th Claws Mail 7/10

Web: www.claws-mail.org
Licence: GPLv3 Version: 3.16
New account setup and documentation are its weaknesses.

5th Mailspring 5/10

Web: https://getmailspring.com
Licence: GPLv3 Version: 1.4.2
Needs to add encryption and spam filtering support pronto!

» ALSO_CONSIDER

As is often the case, we restricted ourselves to graphical applications for this month's *Roundup*, Before we turn to the command-line alternatives, there's still some graphical ones that we must mention. If you like *Claws Mail* then you should also devote attention to *Sylpheed*, since *Claws* was forked from it back in 2005. Interestingly, both tools have a nearly identical feature set. Gnome users can also try *Geary*, but it isn't as feature-packed as *Evolution*.

If you're feeling nostalgic, and remember the days of *Netscape*, *Seamonkey Mail* is a viable alternative and offers all the useful features as the rest, such as spam detection and message filtering. Then there's *Mutt*, which is an incredibly robust mail client. You'll need to install and configure other command-line utilities such as *Procmail* to filter messages. *Alpine* is another console application, like *Mutt*, but is designed to be more beginner friendly. LXF

Roundup

Apache » Nginx » Lighttpd » Openlitespeed » Tomcat.

Kent Elchuk
is an experienced web developer and in his spare time does hydroponic food production.

Web servers

Looking at the options for building, testing and deploying web apps at home or in the office? Well, let's check out some of the best!

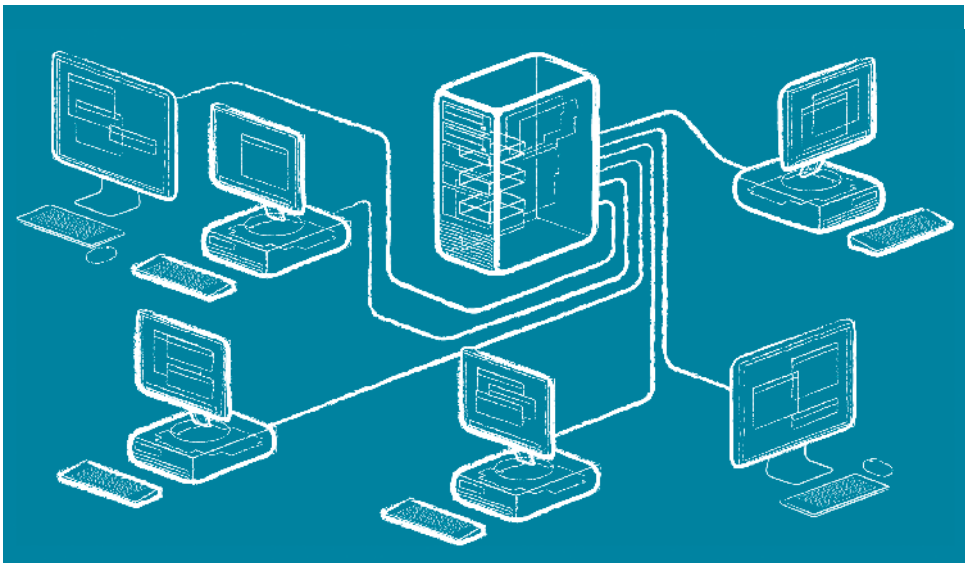
HOW WE TESTED...

Since our Linux box has the ability to build and setup web servers in minutes, our local machine is the device we used in this tutorial to test our servers. In order to complete our testing, all we really need is access to the Internet while installing our building and testing software. Once that has been done, we can manage without it for our testing needs.

Since browsers are the main way users experience server performance these days browser-response times will be a key test, we can test loads and delivery to produce some very useful numbers.

In addition to browsers, we'll use the Apache Benchmark test suite, which delivers some very useful data, this will help us gauge the efficiency of our servers and why some are better in some circumstances.

On top of raw performance we'll also judge each by their ease of installation, availability of support and documentation, the range of devices they support plus general features and any ability to extend the servers beyond their basic feature sets.



Much of the the World Wide Web is used to display readable web pages. From a typical web development point of view, most websites and web applications are built with a combination of web coding languages; like HTML, CSS, JavaScript, PHP, Java and mySQL to name but a few of the most common ones. Going into details a little deeper, web pages can be basic, speed optimised web pages, or complex web applications with lots of bells and whistles like slideshows, data reporting and e-commerce featurest.

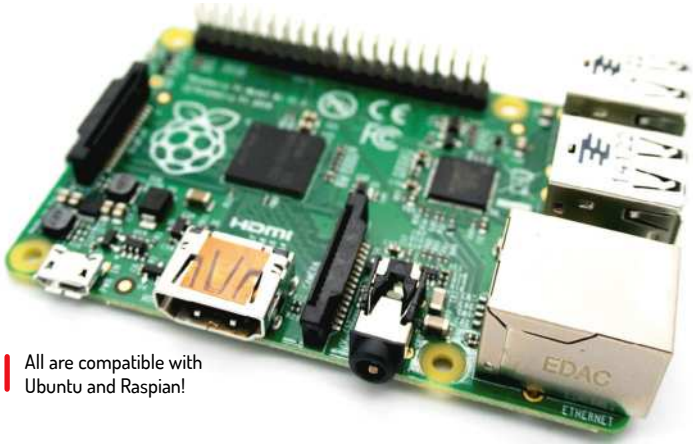
As websites comes in all shapes and sizes, it cannot be stressed enough how essential it is to know which web server delivers the best performance for the content you're dealing with. So, that's just what we will do. We're going to test varying examples of content from the most basic to more elaborate websites, alongside how the servers install, perform and are configured.

> "We're going to test varying examples of content from the most basic to more elaborate websites"

The servers we will examine are Apache, Nginx, Lighttp, Openlitespeed and Tomcat. So, stick around and we'll explore which server suits your needs best. Hopefully you'll discover that the tests here will influence your real-world hosting choices and enable you to host faster serving websites.

Device support

Installing the web servers on Ubuntu and Raspian.



All are compatible with Ubuntu and Raspian!

When running the installations and tests for the various web servers, both Ubuntu and Raspbian were used since both tools are of great use to a typical web developer. In some circumstances, Ubuntu would be a simple install while the latest 2018 version of Raspbian had no packages. Luckily, we used our time to find five packages that did install rather quickly on our machines. In addition to installation issues from simple install commands, installing from source was erroneous in some cases, too. So, with that said, let's take a closer look at what we found in terms of device support when we made our installations.

With Apache, Nginx, lighttpd and Apache Tomcat, all products could be easily installed on Ubuntu and Raspbian with the usual `apt-get ....` command. For example, installing Apache is as easy as typing `apt install apache2`.

Openlitespeed has documentation that claims it can be installed from a repo followed by the typical `apt-get install` command. However, on both Ubuntu and Raspbian that method outright failed. That was a little annoying, but there are other options. After attempting to install Openlitespeed from the source on both devices, that failed too. The `./configure` command could not run as the file was missing from the compressed file that was downloaded. Since they ordered their installation methods from left to right, we had already wasted time getting to our end means. Luckily, on attempt number three, Openlitespeed did install on Ubuntu with the one-line click method, but not on Raspbian. Thus, for testing web servers, this product succeeded at making the bottom of the list in terms of device support in more ways than one.

Besides from Apache and Tomcat, all servers use port 80 for opening web pages. By default, Tomcat uses port 8080. Thus, running Tomcat and using **localhost:8080** will open it. Unfortunately, if we are already using port 8080 for another service, such as Linux Motion, we need to configure it to use another port such 7000.

The good thing about all of the servers is that they install rather quickly, thus, we can use them on our devices almost right away after we execute our commands.

VERDICT

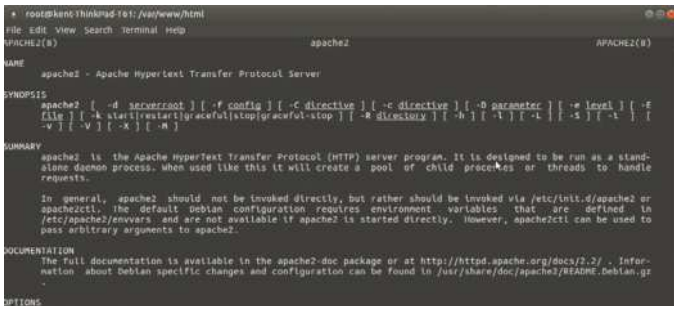
APACHE	**10/10**	**TOMCAT**	**9/10**
NGINX	**10/10**	**OPENLITESPEED**	**6/10**
LIGHTTPD	**10/10**		

It's Openlitespeed that slips here being a lesser-known server than the rest.

Documentation and support

The scoop with man pages and online help

Man pages are limited, but documentation is plentiful.



From a basic installation, only Lighttpd and Apache had man page entries. Meanwhile, all five products tested had man pages online with the exception of Openlitespeed. Although man pages are a great of source of information, there is much better documentation online.

After spending quite some time on each vendor's website during the first-time setup and running, Apache, Tomcat and Nginx seemed to have organised, easy-to-follow information that allowed for an easy installation, modifications and configurations.

Openlitespeed had multiple options and it took an amount of trial and error on both Ubuntu and Raspbian until it worked. Online it does look like a groomed commercial product with a slick GUI that integrated well with WordPress and other software, however it lacks the effort needed for a home web server and testing.

Lighttpd was another lacking web documentation from the source. Luckily, it is easy to setup and use so we avoided a wild goose chase. Finally, and luckily, we have Google to find additional solutions. In fact, we spent most of our time getting the solutions we needed to configure and make everything work perfectly.

With Nginx, Tomcat, Apache and Lighttpd, it seemed there was plenty of good forum questions and answers that could solve our problems. Openlitespeed, although it ranks quite highly for hosting usage on the web, it seemed to have the worst information and answers out there. It was almost as though our questions had no answers and there was very little involvement from home users with this server.

VERDICT

APACHE	**10/10**	**TOMCAT**	**9/10**
NGINX	**10/10**	**OPENLITESPEED**	**5/10**
LIGHTTPD	**7/10**		

Nginx and Apache dominate, and online help is leaps beyond the competition.

Deploying and management

Are the servers easy to set up and use?

Web servers will be installed via the command line. Some installations are very simple one-liners that use `apt-get install` while others need to be installed from source using a few extra commands. Once installed, the servers can be used to quickly display a default web page.

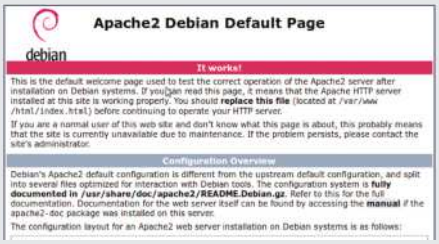
Since we will test our servers with the server-side language PHP and mySQL, we will see how easily they can be used with each server. Looking into details, we will see which software works easiest with basic commands or with the aid of very basic text editing to config files. On top of that, we will look at configuring each server so that it runs more efficiently and helps us get the results we need: a fast-loading website that won't leave potential viewers so frustrated that they abort the site rather than wait for a page to load.

In order to reach this goal, we use the Apache Benchmark tool *ab* to calculate our requests-per-second rate for our web files. This tool can make thousands of requests and simulate concurrent users.

Apache 10/10

During the apache2 installation, we can install mysql-server, PHP, libapache2-mod-php and php-mysql. After this, we can simply write PHP scripts that just work or build a full web app using popular tools such as WordPress, Laravel, or Magento. Apache's ease of installation and performance for testing purposes makes it a great choice for building and testing web applications before they are uploaded and moved to the Internet on some sort of hosting platform; whether that be shared hosting, VPS or dedicated hosting.

Since many web hosts still use Apache, it makes for a consistant, solid, uniform testing system. However, when using Apache, tuning it can be very important since it can be slow if we don't make the best of it. To make testing basic, we have a few tools such as the browser Inspector and the Apache Benchmark tool to help reach our speed goals.



Nginx 9/10

During Nginx installation, we can also install *php-fpm* in order to be able to build PHP apps and test them. However, there is a lot more to it than that. In order to be able to open PHP files in our web browser, we also need to edit the file **/etc/nginx/sites-available/default** and un-comment the stanza that begins with `location ~ \.php$ {`.

After we remove the comments that are blocking the PHP code from running, we can reload Nginx and PHP works just fine. Now, we can run and test any PHP applications whether they are custom scripts, WordPress, Magento, Joomla, Drupal or many others from such a long list. Now, Nginx performs exceptionally well, which is one of the main reasons its popularity has increased over recent years. And since Nginx is so easy to set up and run, it is a wonderful option for third-party web hosting.

Speed and performance

Benchmarking the requests per second.

For our performance tests, we used the browser and the Apache Benchmarking tool. The browser tests were quite simple, we right-clicked on our page, selected 'Network', refreshed the page and clicked the line which showed our URL. Once we clicked the line, we had access to the request and response headers. The headers enable us to see some performance stuff such as the server itself (that is, Nginx, Apache or Lighttpd), Keep-alive, gzip and cache control.

As far as the Apache HTTP Benchmarking tool goes, we used it to test retrieving an image 25,000 times, since this is a good way to gauge a rough of idea of the requests per second that each server was capable of.

To put each through their paces with set them running to fetch over 25,000 requests of a 200Kb JPEG file with 50 concurrent requests (users). The top two servers – both were within the eight-second range – were Nginx and Lighttpd (aka Lighty). Coming in after this bracket, we see Openlightspeed appearing at around 14 seconds, Apache at 19 seconds and, finally, Tomcat just coming in at the rear at a touch over 20 seconds.

Looking into the requests-per-second details, we see Nginx manages around 3,011 with Lighttpd just behind with 2,890. The rest cliump together somewhat further back with Openlitespeed at 1,776, Apache at 1,314 and Tomcat at 1,229. Although these times have quite a spread, the naked eye cannot tell which server is being used since the images and pages load quickly eough with all platforms. Thus, all tools are good for testing web applications on a local machine. When tests were made with an entire URL, like a default WordPress installation, results for speed variance amongst the servers were very similar for the image requests.

VERDICT

APACHE	7/10	TOMCAT	7/10
NGINX	10/10	OPENLITESPEED	8/10
LIGHTTPD	10/10		

Nginx and Lighttpd are just plain faster out of the box.

Lighttpd 10/10

During lighttpd installation, we can install *fast-cgi* and *fastcgi-php* so that we can build PHP apps and test them. But, there is still more to it than that. Lighty installs very quickly and responds almost instantly when a command is typed. In addition, the simple commands do all the hard work for us and both PHP and mySQL database will work just as we expect. However, we must note that if we do not enable and reload Lighty after installing *fast-cgi*, we will receive 403 errors in our browser.

Once Lighty has been installed, it will look for one of the default files (index.html, index.lighttpd.html and index.php) from the **/var/www/html** folder. The quick Lighty setup and the fact that all files and folders reside in the same path as Nginx and Apache make it a gem, too. It's nice to be able to build in one location and test all servers from one spot rather than having files all over the place.

Openlitespeed 8/10

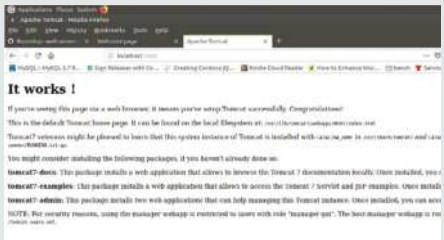
During Openlitespeed installation, an index.html file is created that is used as the default webpage. In addition, PHP works right off the bat since it comes with its own PHP handler called LiteSpeed SAPI PHP. Unlike Nginx, Apache and Lighttpd, the html files are stored at the path **/usr/local/lsws/Example/html/index.html**.

With Openlitespeed, its strengths lies in testing simple HTML and PHP files in the root folder. However, as soon as we want to test a real-world application like WordPress, or the Laravel Framework, its default setup falls flat on its face. If we're working with subdirectories, we need to change the **vhconf.conf** file and add the subfolders manually. Another issue is that our mySQL installations which work seamlessly with Nginx, Apache and Lighttpd, do not work out of the box with Openlitespeed. Any of our previous databases built with PhpMyAdmin cannot be initially used.

Tomcat 9/10

Tomcat installs and works right off the bat. However, like Openlitespeed, Tomcat files, by default, sit in a different location than those for Apache, Nginx and Lighttpd. In fact, they reside in the **/var/lib/tomcat7/webapps/ROOT** directory. In addition, its default port is 8080, unlike other web servers running on port 80.

Unlike all the other servers, Tomcat is designed to deliver Java Servlets and JSPs. Thus, with a Java project, a WAR (web archive) file can be dropped and deployed with Tomcat. Although Tomcat can be used to deliver browser-interpreted files such as HTML, CSS and JavaScript, it can be configured to interpret PHP files as well, although it was not designed for this purpose which makes it more of a specialized tool for programming in Java. Considering Tomcat is Java-based, it does rank fairly well as a popular web server and seems to just work without issues.



Customisation

Adding and enabling modules and custom directory options.

Since most websites will use a server-side language, such as a WordPress installation or basic code to handle a contact form from HTML, let's have a look at how easy that is to implement.

Extending web servers is easier with some than others. However, from a simplicity standpoint, both Apache and Lighttpd make it very easy to check modules and enable them as needed. For example, one such module that is not enabled by default in Apache is mod_cache. Nginx, on the contrary does offer plenty from the first go, but single commands to enable and disable them are not part of the program. With Nginx, adding and configuring modules runs beyond novice usage.

Openlitespeed is not as straightforward as Apache or Lighttpd either, but they both can at least be added to with somewhat simple commands for more intermediate users.

Aside from adding modules to a server, there will come a time when we want to edit the server configuration files themselves or add files to folders for which we can make custom configurations, such as within an **.htaccess** file.

Apache and Tomcat make such configurations very easy since we can add an **.htaccess** file to any folder and make custom rules, browser caching, and so on. Meanwhile, Openlitespeed has methods for which we can migrate our old **.htaccess** from Apache into its own system and use the same rewrite rules, which eases transistion from an old Apache server.

Lighttpd had its own way of creating custom files in its own **lighttpd.conf**. With Nginx and Lighttpd, most custom rules will take place in configuration files.

VERDICT

APACHE	10/10	TOMCAT	9/10
NGINX	8/10	OPENLITESPEED	9/10
LIGHTTPD	8/10		

Apache is very flexible since we can make server rules with a single .htaccess file and enable and disable modules with one-line commands.

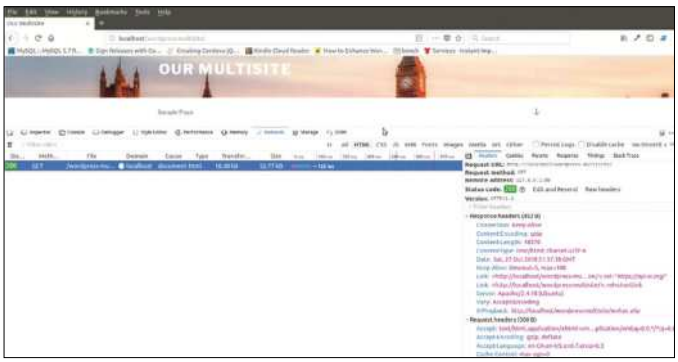
Essential features

Can our web servers run right out of the box – or not?

Out of the box, they all work. However, since we have different and changing needs, we can look at what they come with and how that will work for us. For testing HTML and browser-interpreted code such as CSS and JavaScript they all display static content, by default. But, as soon as we need to add a server-side language like PHP, we not only need that to be installed, but, need the server to communicate and use it effectively. Well, only a couple small tweaks are really required for all packages to make this possible. Once a minute or two is taken to configure PHP, so we can run database applications such as Wordpress, the rest of the details that must be considered are speed optimisation and security. So, let's go through the list and see what we have post installation.

With Apache, the modules that are enabled by default include gems such as core_module, so_module, alias_module, auth_basic_module, deflate_module, dir_module, env_module, php7_module and rewrite_module. In addition to the out-of-the-box modules, Apache has many other that can be enabled with a single command, which includes SSL and mod_expires. Nginx boasts some good ones, too, like fastcgi, ssl, gzip and geoip.

Lighttpd also has an array of modules used by default which are mod_access, mod_alias, mod_compress, mod_redirect and mod_rewrite. These can be found in the **/etc/lighttpd/lighttpd.conf** file. Openlitespeed also works out of the box, even though it lags behind Nginx and Lighttpd. In addition, Openlitespeed requires more intermediate usage in order to customise it further. Tomcat works for simple static files and WAR files and, as such, it's the go-to solution for testing these apps.



Many useful modules are enabled by default and we can also examine the headers with a browser to show features and some data we should hide.

VERDICT

APACHE	10/10	TOMCAT	9/10
NGINX	9/10	OPENLITESPEED	7/10
LIGHTTPD	8/10		

Apache is the only server that a noob can dive into and configure with basic commands. The others are more aimed at intermediate users.

Security features

Can we host a site safely?

When we start our web server and test local pages, we are the only ones accessing our website with port 80. However, as soon as deploy a site online, we need to take security more seriously since our router will forward port 80, so outside users can access the site. Let's do a quick run-through of our web servers and see what changes we need to make.

With a default Apache installation, we should make several changes; like installing the mod_security module and making several changes to the **/etc/apache2/conf-enabled/security.conf** and **/etc/apache2/apache2.conf** files. With the **/etc/apache2/conf-enabled/security.conf** file, we should uncomment `ServerSignature Off` and `ServerTokens Prod`. We can also edit the **apache2.conf** file to turn off directory browsing and limit large requests to protect from denial-of-service attacks.

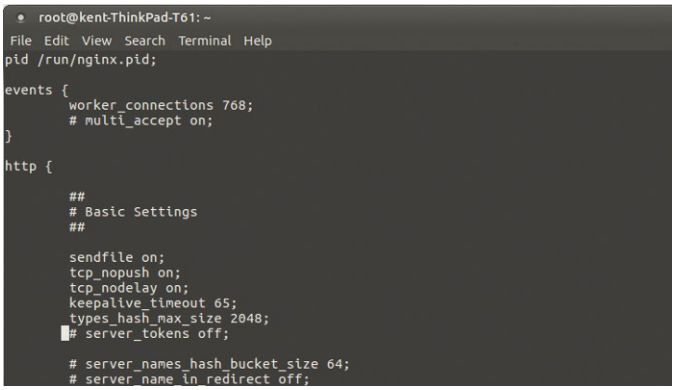
With Nginx, we can open the file **/etc/nginx/nginx.conf** and set `server_tokens off` to hide the server version. In addition, we have the ability to restrict pages by IP so our admin accounts are bullet-proof. We can also chip away at the default config file.

Openlitespeed is very good at controlling files by default. We had to edit the config file to be able to use subfolders and so on. To add more layers of protection, we can install mod_security module and use bandwidth and connection throttling.

With Lighttpd, we can edit the **lighttpd.conf** file to our liking. Thus, we can set numbers for max connections, max keep alive, SSL, max connections per IP and prevent image hijacking.

Tomcat needs quite a few configuration adjustments to deploy a live website. We can hide the server header, enable SSL/TLS and enforce HTTPS. We can also not run Tomcat as root, disable the SHUTDOWN port, disable sending of the X-Powered-By HTTP header, disable Tomcat from displaying directory listings and limit the availability of connector.

In addition, we should change our **php.ini** file with `expose_php = Off`. For further testing, we can always install *wapiti* and other monitoring tools like *Monit* to test and monitor our local websites. Going a step further, we can use Kali Linux and perform penetration testing.



Simple editing and uncommenting lines for a main server config file will help to tighten up security for your website deployment.

VERDICT

APACHE	8/10	TOMCAT	7/10
NGINX	9/10	OPENLITESPEED	7/10
LIGHTTPD	9/10		

Config files for Nginx, Lighttpd and Apache make it easy to add more security.

Web servers
The Verdict

Sites that rank web servers by popularity seem to mirror very closely the verdict we've arrived to testing these web servers, for home and dev use. Like everything in life, every server has their own strengths and weaknesses, but all of them are more than sufficient for our basic purposes, which is to build a development server at home with the option to still use it as our primary web host if we so choose.

If our primary objective is to be up and running with as little tweaking as possible, Apache is a top choice because it just works perfectly out of the box, after a few installation commands and without having to get under the hood–or no one ever got fired for choosing Apache!

Nginx is right behind Apache for simplicity, as long as we make a simple change to the **.conf** file to enable PHP to run. The extra time needed over Apache to do this is really just a minute or two at most. Thus, making it basically a tie, especiallyu so as Nginx also feels like a solid workhorse, and though absolute perforance isn't critical for us, it clearly outpaces the rest. In addition, if we take web server popularity into account, some sources claim that Apache and Nginx account for over 80 per cent of the server market. Thus, this makes both promising for both at-home testing and syncing web files to a remote server in any environment.

Openlitespeed looks like a good choice for the web and seems to have a real commercial edge to it, even if you ignore that it already has a decent amount of the existing web-hosting share. Nevertheless, it was too far off the grid from Apache and Nginx and had a number of time sinkholes to set up and manage, from a home user's or developer's perspective.

Lighttpd is another tool that works well and runs along the same lines as Nginx and Apache. So, using it is a rather simpler experience in comparison to the other two "trailing" server options.

Tomcat is another great tool and is recommended for those that are on the JSP side of things. Like Apache, Tomcat just feels like a tried-and-trusted way of getting a job done without bugs or poor documentation. The take-away from this *Roundup* is that all five web servers here are mature, reliable tools more than capable of meeting every need.

1st Nginx 10/10

Web: **www.nginx.com** **Licence:** **BSD**
Version: **1.15.5**
The fastest optiojn, easy to use with a solid community behind it.

2nd Apache 9/10

Web: **https://httpd.apache.org** **Licence:** **GPL**
Version: **2.4.18**
Easiest for beginners, best support, but overall speed is its weakness.

3rd Tomcat 8/10

Web: **http://tomcat.apache.org** **Licence:** **ASF**
Version: **8**
The go-to option for Java developers serving Java Servlets and JSPs.

4th Lighttpd 8/10

Web: **https://www.lighttpd.net/** **Licence:** **BSD**
Version: **1.4.35**
It is easy to use and delivers good speed.

5th Openlitespeed 6/10

Web: **https://openlitespeed.org/** **Licence:** **GPL**
Version: **1.4.39**
Unorthodox file locations, not as user-friendly and lags at the back.

» ALSO_CONSIDER

When choosing our list of servers, we wanted to compare the most popular web servers, as there are very good reasons they're used; such as reliability over many years.

However, there is a very long list of other web servers that can be used for testing web apps and deploying live websites such as Tengine, Cherokee, IdeaWebServer, Cowboy and Monkey. Although we were hoping to find some gold in the smaller players, our search came up empty.

Almost all of the others we tried were not worth the effort. In many cases, vendor websites looked abandoned and instructions failed. Often, the packages did not exist and files were missing when building from source. When a package did work, it would have taken too long to configure to be useful.

We are happy to have run samples and tests on these alternative options, but there's little reason for trying options outside of the core servers in this *Roundup*.

Roundup

Clonezilla » Deepin Clone » FOG Project » G4L » WereSync

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

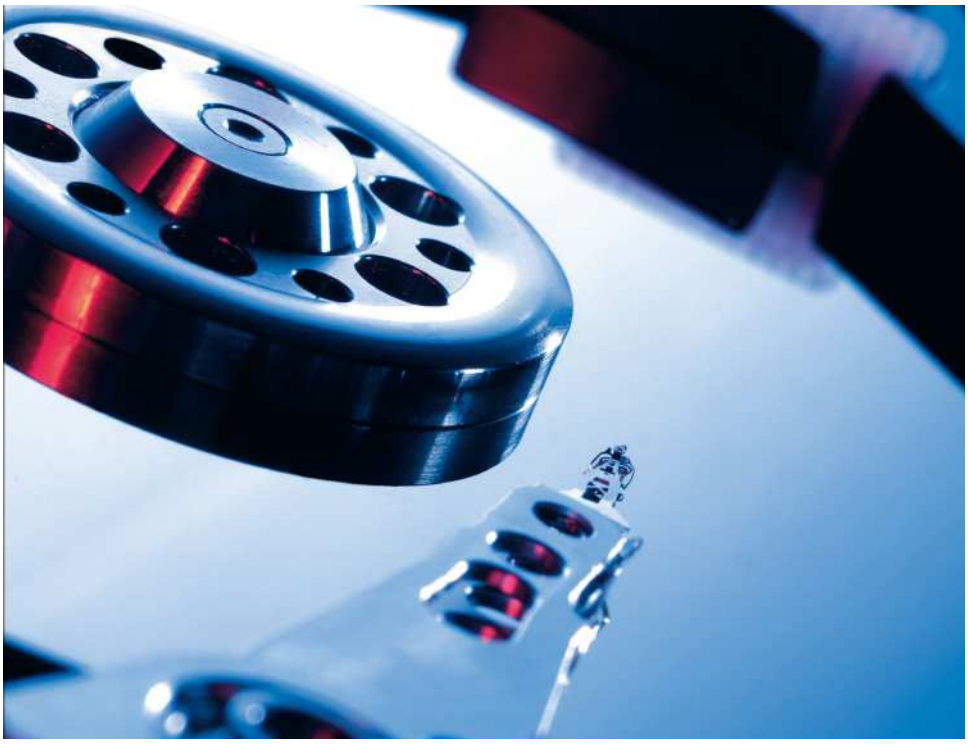
Disk cloning tools

Need to provision a lab full of PCs? The ever lethargic **Shashank Sharma** tests the best tools to get the job done without breaking into a sweat.

HOW WE TESTED...

A disk cloning tool requires quite a bit of hardware to be tested effectively, particularly hard disks. This is why virtual hardware is ideal for testing their capabilities. For this *Roundup* we're using several virtual machines with multiple hard disks of various sizes. We'll use them to image a small disk to a large one, which is one of the most popular uses of a disk cloning tool.

Secondly, we'll clone a large disk with lots of empty space to a smaller one with enough room to accommodate the used sectors of the larger disk. The *Clonezilla*, *G4L* and *FOG Project* tools have been tested on Manjaro, Fedora and Ubuntu installations. *WereSync* was installed on Manjaro and *Deepin Clone* was tested atop a Deepin 15.9 installation for best results. The network-based *FOG Project* was installed on a Ubuntu Server installation as per its documentation.



Managing a network of computers is an involved process. The constant barrage of repetitive tasks such as running checks, troubleshooting errors, replacing dead hard disks and doing fresh installations over and over again can sap the energy out of any system admin, irrespective of the size of your realm. Even before you can tackle the problem of actively monitoring the machines, you have to install an operating system on each of them. This is a time-consuming task even for a small network with, say, 10 computers.

Computer cloning involves setting up the operating system, drivers, software and data on one computer, and then automatically replicating the same setup on other computers. The technique, also known as ghosting or imaging, is used by system admins for rolling out multiple identical machines over the network.

The cloning tools on test here are a blessing for harried admins who want to put their feet up every now and then. With these tools you can image and clone machines without breaking into a sweat. We'll look at tools that, besides cloning, will also help take the pain out of everyday admin tasks such as installing software, and can scale up to work over large networks and multiple locations.

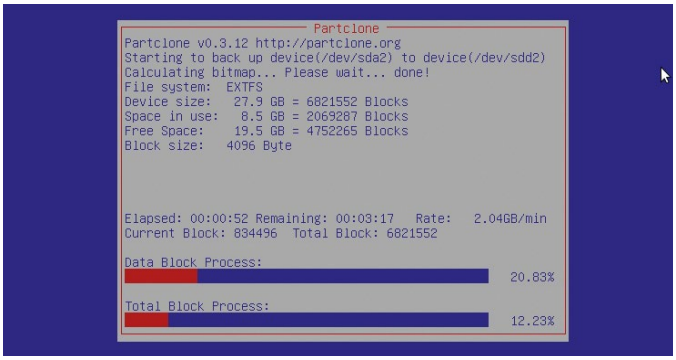
Cloning flexibility

Do they offer enough options to meet your cloning requirements?

Clonezilla offers a lot of control over the cloning process. You can use it to clone individual partitions or complete disks, or to clone the disk or partitions to an image – which can then be saved to another locally attached disk, or even a removable drive. Rather than cloning to an image, *Clonezilla* can also directly clone the disk or a partition to another disk or partition. Although this process worked for us when cloning entire disks, we wouldn't recommend it and would rather suggest you go via the imaging route, which is a lot safer and offers far more flexibility than direct disk-to-disk cloning.

Much like *Clonezilla*, *G4L* has a very verbose interface, which gives you several options to help fine-tune settings when creating and restoring images. Like *Clonezilla*, you can use *G4L* in separate modes, with the RAW mode that can clone all types of disks and partitions being the most useful. You can save the images locally or across the network via FTP, SSH, SMB or NFS. For the impatient, *G4L* also offers the option to directly clone a drive without imaging it first.

Unlike the previous programs, *FOG Project* doesn't work as a Live CD and thus has a different cloning mechanism. When you use *FOG Project* to image a computer, it offers plenty of options, with various fields to describe the host images. It can also arrange the images into groups for easier management. There are also several options to schedule the imaging process, if you don't wish to image them immediately. *WereSync*, meanwhile, is a Python script that has a graphical interface and is designed for regular users. The utility only offers direct drive-to-drive cloning, instead of the more popular and safe drive-to-image cloning option offered by the others.



Direct disk-to-disk cloning options are faster than going via the image route, but come with certain amount of risk and don't offer the same flexibility.

Deepin Clone's primary option is also a direct clone from one disk to another, although it does offer the option to clone and restore via an image too. The app enables you to clone complete disks as well as individual partitions. Note however that it cannot clone mounted partitions, nor can it save images on the same partition or disk it is cloning. It also doesn't offer any options to compress, split or encrypt the cloned image.

VERDICT

CLONEZILLA	**8/10**	**G4L**	**8/10**
DEEPIN CLONE	**5/10**	**WERESYNC**	**6/10**
FOG PROJECT	**8/10**		

They all offer enough dexterity for imaging a disk or partition.

Other custom parameters

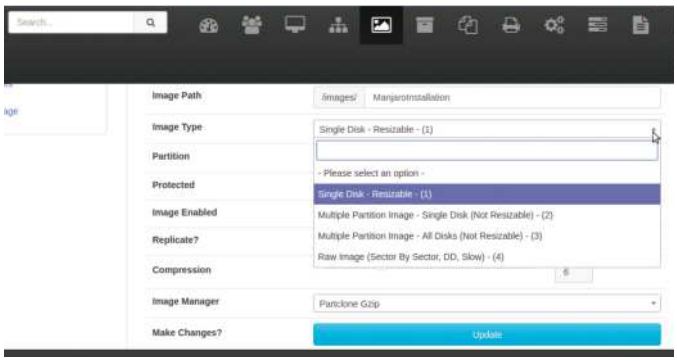
Do they offer any supplementary options to assist with cloning?

Apart from presenting you with a list of disks and partitions to clone, *Clonezilla* also gives you options to compress them using multiple compression algorithms, and even optionally encrypt the cloned images for added security. A major step in the cloning process is to select the storage location of the cloned images, and *Clonezilla* enables you to save them to a locally attached disk or on another computer via the network.

G4L has a huge list of menus that help you with each step. You can earmark the partitions or disks you want to image, name the image, select the compression algorithm and more. *G4L* will help you transfer images to remote destinations on the network, and you can also split them up for easier storage on smaller drives.

You can create several different tasks for any of the hosts in *FOG Project*'s repository. You can run the Debug task which boots a Linux image to a *Bash* prompt for fixing any boot errors. You can also create a task to remote wipe hosts, to recover files with *TestDisk*, or to scan for viruses with *ClamAV*. The *FOG Project* server can also install and manage printers on the network. *FOG Project* can also install and uninstall apps via snapins.

Both *WereSync* and *Deepin Clone* offer very limited functionality and are less flexible than the others.



With FOG Project you can also track a user's access to their computers, and shut down the machines after a specified period of inactivity.

VERDICT

CLONEZILLA	**8/10**	**G4L**	**8/10**
DEEPIN CLONE	**4/10**	**WERESYNC**	**5/10**
FOG PROJECT	**9/10**		

The app-based options fare poorly in terms of imaging dexterity.

Usability

What makes them stand out?

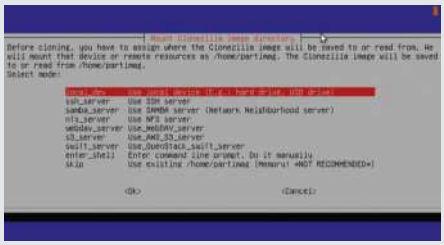
Unlike desktop apps, disk cloning tools are administrative software and you'd expect them to sacrifice a certain amount of user-friendliness in order to make room for advanced functions and features. That being said, while we don't expect these tools to have point-and-click usability, a certain amount of intuitiveness will surely make them more approachable.

We are in no way expecting these tools to cater to a non-technical audience that doesn't know what they are getting into, but having a logically laid-out interface with a clear path of progression will go a long way to getting the most out of the tools. Ease of deployment and reasonable defaults are the icing on the cake, and while they aren't all that critical to selecting a good administrative tool, they can't really hurt its chances.

Clonezilla 8/10

Clonezilla is one of the most recommended tools on Linux forums for imaging an old disk to a bigger, new disk thanks to its ncurses-based frontend, which is intuitive (*Really?–Ed*) to navigate. In the background, *Clonezilla* uses a set of scripts with several open source disk utilities such as *Partimage*, *ntfsclone*, *Partclone* and *dd* to help you duplicate individual partitions and complete disks.

The menus offer sensible defaults so you'll likely go with the recommended options most of the time. Just make sure that the partition to be cloned isn't mounted. Also, despite its unglamorous text-based interface, *Clonezilla* isn't intimidating because of its very verbose screens. Each step is dotted with relevant information to help you make an informed choice. Still, *Clonezilla* isn't something to be approached without adequate knowledge and experience.



Deepin Clone 8/10

If there's one area where *Deepin Clone* trumps the rest of the competition, it's usability: the app really is as simple as they come.

On the first screen, it asks you to select whether you wish to clone a complete or individual partitions. Depending on the selection, it'll bring up another set of options to help you narrow down the disk/partition you want.

On the following screen you're shown a list of disks or partitions. Unlike some of the other options on test here, if you've selected the option to clone partitions, the app won't let you clone multiple partitions in one go. Also, its default mode of cloning is a direct disk-to-disk or partition-to-partition, which is rather unsafe and something we'd never recommend doing. In addition, to use *Deepin Clone* to image the current installation, you'll have to run it via *Deepin Recovery*.

Supported filesystems

Will they work across all machines in your network?

Clonezilla supports a wide number of filesystems, from popular ones like ext2/3/4, ReiserFS4, Btrfs, FAT16/32 and NTFS to specialised ones like XFS, JFS and UFS. But that doesn't mean it's totally unusable for unsupported filesystems. For those it supports, *Clonezilla* images only the used blocks for greater restore functionality. If it runs into a filesystem it doesn't understand, *Clonezilla* uses *dd* to image the contents as-is, and can restore these easily to equal-sized disks.

Much like *Clonezilla*, *G4L* works in two modes. There's the filesystem level where it uses utilities such as *NTFSclone* and *FSArchiver* to directly clone supported filesystems, including virtually all the popular ones. If you work with esoteric filesystems or need to clone entire disks, you can use *G4L* in RAW mode to copy all bits of any file system.

FOG Project uses both *PartImage* and *PartClone* in the background, so it supports all the popular filesystems. The Imaging section also enables you to create a sector-by-sector image of the drive using *dd*.

In contrast, there isn't much information on the internet about the way that *Deepin Clone* works, and it doesn't offer any configurable parameters that would hint at the filesystems it supports. In our tests, however, it worked on both ext4 and NTFS partitions, so there's that.

In contrast to its peers, *WereSync* uses *rsync* to copy the contents of the drive, and can image the contents of GPT, MBR and LVM drives. According to the official documentation, you can even force the tool to copy the contents of a drive to a file by specifying the IMG extension type.

VERDICT

CLONEZILLA	9/10	G4L	9/10
DEEPIN CLONE	5/10	WERESYNC	5/10
FOG PROJECT	9/10		

All these tools give you decent mileage so long as you're working with one of the popular filesystems.

FOG Project 8/10

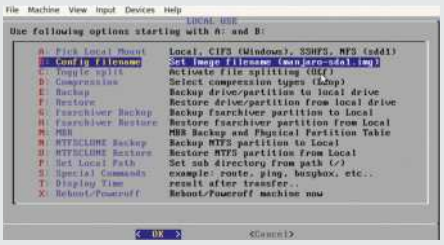
FOG Project has a much more involved installation process than the other solutions here. Unlike *Clonezilla* and *G4L*, you first need to set up an imaging server. While the initial deployment might be time-consuming, *FOG Project*'s web interface makes the effort worthwhile since it can image and deploy multiple computers simultaneously with ease.

FOG Project's web interface is fairly intuitive for anyone who is familiar with the intricacies of the imaging process. The menus are arranged in a logical fashion and first time users should have no issues navigating it. The setup is scalable as well, and you can set up dedicated storage nodes which can help take the load off the main *FOG Project* server, if you have to host and deploy images to a large number of computers. You can use the web interface to manage all imaged computers as well as their cloned images.

G4L 8/10

As with *Clonezilla*, the interface of the *G4L* Live CD is text-based and written in ncurses. This one, however, is much more verbose. The process of creating and restoring images takes you through several wizards, each of which is loaded with quite a number of steps. Unlike *Clonezilla*, *G4L* ships with four different kernels so you'll first have to select the one based on your hardware. If you're using *G4L* on an older machine, use an older kernel for greater compatibility.

Upon booting, the Live environment will first display some basic usage information spread across multiple pages. Advanced users can then either issue a cloning or restoration command directly, or use the interface to construct the command using the various menus. As you step through each screen, *G4L* brings up another screen asking for more relevant options to complete the task.



WereSync 6/10

Installing *WereSync* is fairly straightforward since you can grab it from Python's *PIP* package manager with a single command. You can then fire up the graphical interface which is fairly intuitive, although there are some options that don't immediately make sense.

A quick scroll through the documentation is advisable to familiarise yourself with the peculiarities of the tool. Some of the options also have a brief explanation built into the app itself. Despite these avenues of help, it's difficult to differentiate between options that are mandatory and optional, and the impact that their inclusion or exclusion will have on the cloning process.

Also disappointing is the lack of meaningful errors. By default the app is configured such that once begun, all cloning tasks end with a success message, even when the process has failed.

Help and support

Where do you go if you run into trouble?

Clonezilla hosts a pretty comprehensive FAQ on its website, which you should scroll through before experimenting with the tool. You can also engage directly with *Clonezilla*'s developer via the official forum boards, where he is fairly active. New users can also find getting started information littered all over the web, including on YouTube.

The primary source of information about *G4L* is its verbose interface. The project is hosted on SourceForge and besides the built-in help, there's little else on its website. *G4L* does have an active forum where you can engage with the developer, who again is fairly active. Again, just like *Clonezilla*, *G4L* is fairly well-covered and you can find lots of getting started information and tutorials all over the internet.

New users will be well taken care of by *FOG Project*. The project has extensive documentation on its website including a wiki and a detailed FAQ. There's also an installation guide with distro-specific installation notes. If you run into any trouble with your *FOG Project* installation you can get help from its very active forum boards, and can even find its developers and seasoned users via its IRC channel.

Although it has a very intuitive interface, as we've said there isn't much information available about *Deepin Clone*. It doesn't have any dedicated avenues of support besides the ones for the main distro.

There are two parts to the *WereSync* tool, a command-line utility and a graphical one. The project's webpage has adequate information to orientate new users with both the utilities, including an in-depth explanation of all its CLI parameters.

VERDICT

CLONEZILLA	9/10	G4L	8/10
DEEPIN CLONE	5/10	WERESYNC	5/10
FOG PROJECT	9/10		

While the mature tools don't always have good documentation of their own, you can find help elsewhere on the internet.

Network use

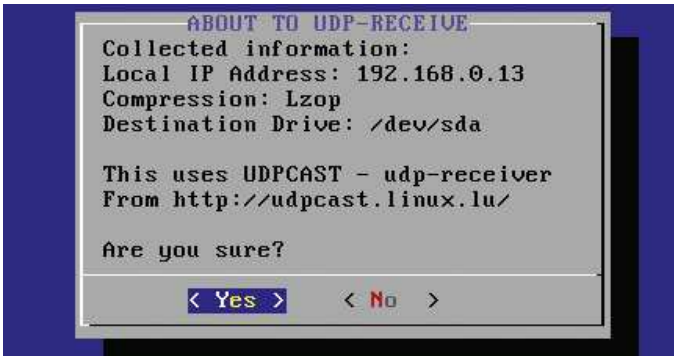
Are these tools good for only imaging local disks and partitions?

You can use *Clonezilla* for multicast cloning via two available options. *Clonezilla Server Edition* requires setting up a Diskless Remote Boot in Linux (DRBL) server to broadcast images across the network. On a smaller network, you can use the *Clonezilla Lite Server* mode on the bootable CD to set up a temporary server.

You use *Clonezilla* to save the cloned image of a disk or individual partitions over the network. For this it can establish a connection to the remote machine via SSH, or through an SMB, NFS or WebDAV server.

Similarly, in addition to creating and restoring local images, *G4L* also has impressive network capabilities. You can use it to backup/restore and image to/from a remote machine on the network via various mechanisms, including FTP and UDPcast. To use the latter, configure the Live environment to set up one machine as a multicast server, which then broadcasts images stored locally or elsewhere on the network.

Unlike the other tools on test here, *FOG Project* is built from the ground up to work over the network. It works best when it doubles up as the DHCP and PXE server. All of *FOG Project*'s imaging tasks, including capturing and deployment, are handled remotely over the network. Unlike the other solutions it forces the use of a lot more network services such as DHCP and PXE servers, which are optional with the other tools. By contrast, neither *Deepin Clone* nor *WereSync* offer any network awareness of their own. With *Deepin Clone*, though, you could obviously mount a remote computer in the local filesystem and point to it as the destination for saving the image of a local disk/partition.



You can use the G4L Live CD to turn remote computers into receivers that'll pull the broadcasted images and restore them over the network.

VERDICT

CLONEZILLA	8/10	G4L	8/10
DEEPIN CLONE	5/10	WERESYNC	1/10
FOG PROJECT	9/10		

FOG Project is designed from the ground up to reside on the network, while Clonezilla and G4L can be used to work across the network as well.

Deploying images

Putting the cloned images to use.

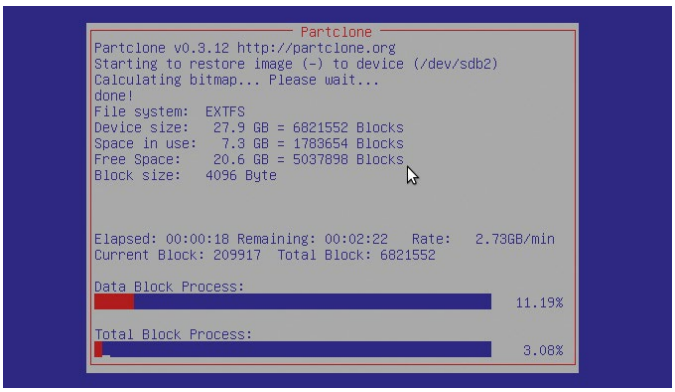
We wouldn't fault you for thinking that the process for restoration would be the reverse of the imaging process for each of the tools. But *Deepin Clone* and *WereSync* will surprise you in this respect.

Clonezilla has a very straightforward process for restoring a partition or an entire disk from a cloned image. You begin by first selecting the repository that houses the cloned images. Next, *Clonezilla* displays a list of all the unmounted partitions to which you can restore the image.

After you've made your selection, *Clonezilla* wipes the partition and creates a new partition table on the disk, so be careful with this. Also keep in mind that the destination partition needs to be equal to or larger than the source, as it can't restore an image to smaller disks.

When restoring with *G4L*, you must point the tool to the saved image, followed by the location of the disk or partitions where you want to restore the image. During our tests, while *G4L* certainly began restoring a cloned image from a larger disk to a smaller one, the process never completed successfully. On the other hand, attempts at restoring images of a smaller disk to a larger one went smoothly.

With *FOG Project*, you first need to register the target machine with the server and then associate a cloned image with it. To do the actual rollout, you'll have to create a deploy image task before the image is rolled over to the machine via PXE. The same process works irrespective of whether you wish to deploy to a single computer or multiple ones.



All tools can perfectly restore a smaller image to a larger disk, but only WereSync promises to restore a larger image to a smaller disk.

To restore an image with *Deepin Clone*, all you have to do is point the app to the image file along with the disk or partition you want to restore it to. As with its cloning process, the restoration doesn't offer any configurable options.

By default, *WereSync* directly clones a drive to another, so there's no option for restoration. However, you can force the app to image a drive by changing the destination to an IMG file. To restore from this image, you'll have to reverse the fields and specify the IMG file as the source.

VERDICT

CLONEZILLA	8/10	G4L	8/10
DEEPIN CLONE	5/10	WERESYNC	5/10
FOG PROJECT	9/10		

They're all well-matched, with the exception of WereSync and Deepin Clone.

Disk cloning tools
The Verdict

With their varied ways of working, you can find a situation that's best suited to each one of the tools featured here. For instance, *Deepin Clone* is an ideal intuitive solution for home users, which just works. But while the app is available for non-Deepin distros such as Arch, we don't feel comfortable using it in any other distro.

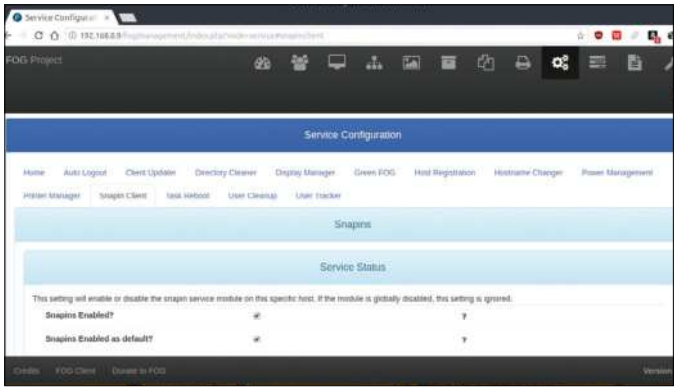
WereSync is based on the venerable *rsync* utility and is the only one that can clone a larger disk to a smaller one. It didn't make the podium on account of its direct disk-to-disk cloning mechanism, which we dislike. Coupled with its limited dexterity and sparse documentation, the program manages to reach the fourth spot at best.

While advanced users will probably have no issues working through *G4L*'s interface, first-time users will surely feel inundated by all the options at their disposal. Critical administrative software, such as disk cloning tools, require a certain degree of comforting familiarity, and new users will take quite some time to get there with *G4L*.

All things considered, *Clonezilla* and *FOG Project* are the only viable options that also scale well. We can easily recommend them to individual users for use on their home network, as well as to admins of larger networks.

While both do similar tasks, they go about it very differently. The most visible differences between the two solutions are their user interfaces and the fact that *Clonezilla* runs from a Live CD, while *FOG Project* requires setting up a server. *FOG Project* is also more comprehensive in that it can perform several tasks in addition to imaging, such as installing apps remotely in Windows installation via a feature called snap-ins.

If your machines aren't on a network, you'll have no option but to use *Clonezilla*. Both solutions are on an equal footing in a networked environment, but *Clonezilla*'s network options make the most sense in an environment like a library, where the remote clients don't require much administration once they've been imaged. If you need to look after machines once they've been cloned, there's no better solution than *FOG Project*, as its post-imaging options are excellent.



1st FOG Project 9/10

Web: https://fogproject.org Licence: GPL v3
Version: 1.5.5
A comprehensive imaging solution that works across networks of all sizes.

2nd Clonezilla 8/10

Web: https://clonezilla.org Licence: GPL v2
Version: 2.6.0-37
A feature-rich tool that works with all the usual cloning use-cases.

3rd G4L 8/10

Web: https://sourceforge.net/projects/g4l Licence: GPL v2
Version: 0.55
Similar to Clonezilla, but loses out because of its cumbersome UI.

4th Deepin Clone 5/10

Web: www.deepin.org/en/original/deepin-clone
Licence: GPL v3 Version: 1.1.0
Works as advertised but currently only caters to users of the Deepin distro.

5th WereSync 4/10

Web: https://github.com/DonyorM/WereSync
Licence: Apache License 2.0 Version: 1.0
Can clone disks, but its design and backend restricts its use cases.

» ALSO CONSIDER

Two of the tools featured in **LXF** in the past, but not included this time around due to them not showing any signs of life, are *Redo Backup & Recovery* and *Mondo Rescue*. The former is by far the easiest bare-metal cloning option we've run into. Its downside, however, is that it takes away a lot of the control over the cloning process in lieu of convenience.

The only real alternatives for cloning disks besides the ones we've already covered in the *Roundup* are the command-line ones that do the grunt work in the background. The two that are really worth mentioning are *Partclone* and *FSarchiver*. They both support a large number of filesystem formats and can quickly roll disks and partitions into movable archives.

The real CLI veterans, however, don't need anything else besides the venerable *dd* tool, which can be used to effortlessly clone a disk or a partition with a single command. You'll need to know what you're doing!

Image credit: Jon Ball

OPEN SOCIAL NETWORKS

Tired of Facebook knowing him better than he knows himself, **Jonni Bidwell** goes in search of a more open and less creepy open social network.

Arguments about whether having such a rich, and carefully curated, online persona (which may not be a particularly good reflection of who we are in meatspace) is a healthy thing that we can save for another day. But as readers of this fine magazine, we can all agree that if we're going to put our media and ramblings online, then things would be a whole lot better if the sites that handle them should be open source, and have clear policies about how that data is used.

Better yet, we should be able to host our own instances of these services. This isn't as far fetched as it seems. Decentralised and federated architectures enable different instances to connect with one another, so an account on each platform is all that's necessary. Not like the old days of following so many blogs across MySpace, Bebo and Geocities, as well as that one guy who hosted his own static site and wrote screeds lamenting the monetisation of the web, and the perils of using 'free' yet closed platforms.

Maybe his message would be heard but for our being distracted by all the cat photos, "influencers" and the rise of meme culture on all those proprietary social networks...

Social networks have become an increasingly dominant force in our online lives. The major players here – Facebook, Twitter, Instagram, Tumblr and let's be generous and even include Google+ – all have different platform features, but they have some key things in common.

Thing One is that they're all centralised services with huge userbases, run by single companies with commercial interests. Thing Two is that they're able to provide their services for free because they make money from advertising. Thing Three is that the reason they can make so much money from advertising is because they're able to supply highly targeted advertising to the highest bidder, because they hold a considerable amount of information on their users. None of these are particularly encouraging, yet somehow a goodly portion of the Internet-accessing world holds accounts on one or more of these services.

Moving away from a centralised service like Twitter, to something decentralised like Mastodon is an empowering process. But it can also be a confusing one. Anyone can host a Mastodon instance (we'll show you how in the coming pages) and lots of people do. So there are thousands to choose from.

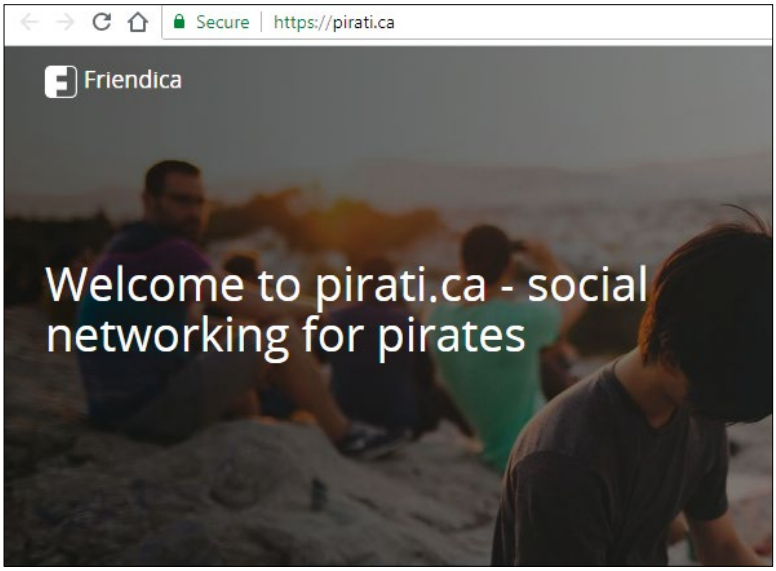
The main Mastodon

People's first question, then, is often, "Which is the main instance?" This is easy to laugh off and respond to with some sarcastic guidance to the dictionary definition of "decentralised". Confusingly though, there's a sort-of flagship instance at **https://mastodon.social**, where a number of Mastodon celebrities, influencers and tech journalists (*what now?–Ed*) can be found. This instance is administered by Mastodon developer Eugen Rochko and periodically is closed to new registrations.

Even if it's not though, it's worth looking at other instances. There's even a handy site (**http://instances.social**) that will help you find one based on your interests, language and tolerance of movie spoilers. Mastodon instances are communities within themselves, vaguely akin to forums of old or subreddits of today. Some are close knit and small, while others are large and loosely connected. Some have strict codes of conduct, some tolerate things which may upset some people's sensitivities, others are just plain weird.

Once you're signed up to an instance, you can view the Local timeline, which are all the posts (toots) of everyone on that instance (in chronological order with no algorithmic voodoo re-ordering or promoted postings). For small communities this is great; for larger ones this can be like drinking from a fire hose – one that serves a variety of potable and non-potable liquids. So, just like on Twitter (pejoratively referred to as the birdsite on Mastodon), you can follow individuals on an instance, and then view a Home timeline consisting of those accounts.

The neat thing about Mastodon is that having an account on a given instance enables you to follow accounts on other instances. This is what's known in the business as federation, which as well as sounding fairly Trekkie is all the rage right now. It's also nothing new. Think about how email works: different providers can talk to each other, and even people signed up to a particular provider may enjoy some extra closeness (instant delivery, less spam filtering, integrated features specific to that provider).



Friendica is federated via the pump.io protocol and has at least one node dedicated entirely to pirates. Yarr.

A collection of federated components is known as a fediverse, so in the context of Mastodon this term refers to all the Mastodon instances. Besides an instance's Local timeline, you can view its Federated timeline. This consists of toots from all the accounts followed by members of that instance. This is a good way to find accounts on other instances to follow. More precisely, short of knowing someone's username, it's currently the only way to find outsiders to follow. If you do know someone's username, then you can just type it into the search box on any instance and follow straight from

» MASTODON AIN'T TWITTER

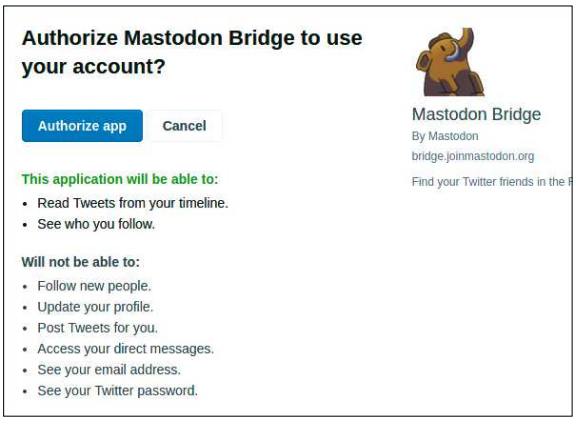
We mentioned earlier not being able to search for individuals by name. This is a deliberate measure designed to reduce harassment, protect privacy and generally make the Mastodon fediverse a more wholesome place than the birdsite (*oh, look at you all down with the lingo – Ed*). In fact, a Mastodon instance is only searchable through the use of hashtags, so if you want your toot to be found you'll need to, in the immortal words of UK Home Secretary Amber Rudd, understand the necessary hashtags. Mastodon communities may be more or less tolerant of certain kinds of speech, but instances hosting (potentially) offensive content can be blocked or silenced by other instances.

Other key differences to Twitter include a 500-character limit on toots, so that discussions can be more meaningful, and much more flexibility about who can see one's toots: they can be visible to specific users, all followers, hidden from public timelines or public. Furthermore, there's a distinctly tusks up attitude towards advertising. It's not that corporate presences are banned outright, but the emphasis is very much on each account having a single, real person behind it (even bots have their masters). Excepting a Mastodon administrator modifying the codebase (which would only affect their instance), there's no opportunity for paid advertising or data mining with Mastodon – it's designed to be about human communication, not making money.

there. If an instance's Local timeline is information overload (or even if it isn't), then it's Federated timeline will be even more so.

Where you @?

Like Twitter, usernames on a given instance are prefixed by @. Unlike Twitter, they're also postfixed with another @ followed by the instance domain. For example, Eugen Rochko is **@Gargron@mastodon.social** and Cory Doctorow (our favourite visionary, author and DRM-freedom fighter) is **@doctorow@mamot.fr**. These look odd, but then all these new-fangled hashtags and handles and URI schema confuse us. If you're referring to someone on the same instance as you in a toot, then the instance name can be omitted. The decentralised nature of the network means someone on one instance can have the same handle (the part of the username before the second @) as someone on another. This allows for some degree of impersonation, which should be reported to the administrator, but no more so than the imitation of usernames on other platforms.



The Mastodon bridge site https://bridge.joinmastodon.org enables you to find your Twitter friends on Mastodon.

» BUCKING THE OSTATUS QUO

Back when the idea of a decentralised social network was in its genesis (around 2009), a protocol family was designed that would cater to this brave new world. That family was called OStatus. OStatus defines protocols that permit de rigueur social networking activities:

Atom feeds Users can publish feeds summarising their recent posts

Activity streams A JSON-based format for encoding activity and event metadata, such as 'liking' a post

WebSub Formerly PubSubHubbub (PuSH) a realtime notification protocol for streams, feeds and the like

Salmon A message exchange protocol for unifying commentary on posts and articles. Provides a single comment thread where the same source is referenced in different places. So-called because salmon swim upstream.

WebFinger A discovery protocol for entities identified by a URI. These may be people, in which case the protocol is functionally (but not at all structurally) similar to ARPANET's Finger protocol.

Much of OStatus's design has been superseded by modern alternatives. OStatus doesn't support private posts natively, which means that platforms that do (such as Mastodon) would have to use extra attributes to federate private messages, and there's no guarantee that other platforms would respect those attributes.

But fear not, because since version 1.6 (released in September 2017) Mastodon implements the new and more flexible ActivityPub protocol, which, knowing now what we didn't then, efficiently provides everything a distributed social network needs. Furthermore, it is extensible enough to be able to provide the things it will need in the future and already can be used for much more than just microblogging – it's at the heart of PeerTube for example. Meantime, Mastodon will support OStatus as a fallback until the 2.0 release.

For a more detailed analysis and untangling of these protocols, we recommend studying JB Crawford's excellent treatment, which can be found at **https://lwn.net/Articles/741218**.

Mastodon may have received a lot of coverage for being a viable Twitter alternative, but it's not the only one. Free Social, a fork of **Status.net**, garnered much interest, and in a way still does, but has since been brought into the fold of yet another microblogging service, GNU Social. This uses the OStatus family of protocols to federate not only with other GNU Social instances, but anything else that implements OStatus, including (for now, see the box, *below left*) Mastodon and Friendica. GNU Social is the only social network where you'll find one Richard M Stallman, albeit in only in the form of a script summarising his website updates.

Instance Mastodon

If you want to run a Mastodon instance that other people (or even just yourself) can connect to, you'll need a server to run it on. This may be a Raspberry Pi running in your basement, or a VPS (you can rent a small one for as little as $5/month), or anything really. It should be accessible 24/7, so running it from your home computer that you turn off every night or regularly play resource-intensive games on is probably a bad idea.

Furthermore, if you do choose to run Mastodon at home, you'll need to configure your router to forward traffic to the server, either using NAT (for IPv4 addresses) or by making an exception in the firewall (most home routers, with good reason, block all incoming IPv6 connection attempts). How to do this depends on the router involved. You'll also want to set up a domain name for your server (nobody likes accessing these thing via IP address), which you can do using either a commercial registrar, or a free service (such as DuckDNS, which issues subdomains of the form example.duckdns.org). Novelty domains (ones ending in .social, .xyz, .rocks or other nu-school suffixes) are popular for Mastodon instances. If you're just testing, you can use a fictitious domain name and place an entry in **/etc/hosts**.

Like so many web applications, you won't find Mastodon in your distro's repos and it's best installed straight from source. Getting everything set up is a lengthy process. We need to set up Node.js, Ruby a PostgreSQL database and a webserver (and associated certificates). You may prefer to use a Docker image. Prebuilt images are available at **https://hub.docker.com/r/tootsuite/mastodon** , and they can be used as-is or customised to suit your needs. There are also turn-key, paid-for offerings available on Heroku and Scalify. Have a read of **https://github.com/tootsuite/documentation#running-mastodon** to study the available options. We'll detail the standalone installation process now, which is based on Ubuntu 18.04 Server, but apart from a few cosmetic differences the process is similar on other distros, and it's identical if you're using desktop Ubuntu.

First, on newer installs (those based on the 18.04.1 point release) it's necessary to enable the Multiverse and Restricted repositories:

```
$ sudo add-apt-repository multiverse
$ sudo add-apt-repository restricted
```

We also need to add the node.js 8 and Yarn repos, which uses the slightly unsavoury but undeniably convenient practice of piping curl output to bash:

```
$ curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -
$ curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
$ echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
$ sudo apt update
```

Now we are ready to install the dependencies, of which there are many (you may want to copy and paste this command the Tootsuite GitHub pages linked previously – just follow the Standalone Installation link):

```
$ sudo apt -y install imagemagick ffmpeg libpq-dev libxml2-dev libxslt1-dev file git-core g++ libprotobuf-dev protobuf-compiler pkg-config nodejs gcc autoconf bison build-essential libssl-dev libyaml-dev libreadline6-dev zlib1g-dev libncurses5-dev libffi-dev libgdbm5 libgdbm-dev nginx redis-server redis-tools postgresql postgresql-contrib certbot yarn libidn11-dev libicu-dev
```

Now we'll set up a Mastodon user and build a Ruby environment, as follows:

```
$ sudo adduser mastodon
$ sudo su - mastodon
$ git clone https://github.com/rbenv/rbenv.git ~/.rbenv
$ cd ~/.rbenv && src/configure && make -C src
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc
```

» SOCIAL NETWORK OPEN SOURCERY

We've focused on Mastodon here, but there are all kinds of other open source, decentralised alternatives to popular social media platforms. PeerTube is a video-sharing platform that uses WebTorrent technology to ease the burden on servers (again, anyone can host a server and servers can federate with each other). When a user views a video they also share it with other users, creating a sort of positive feedback loop for popular videos.

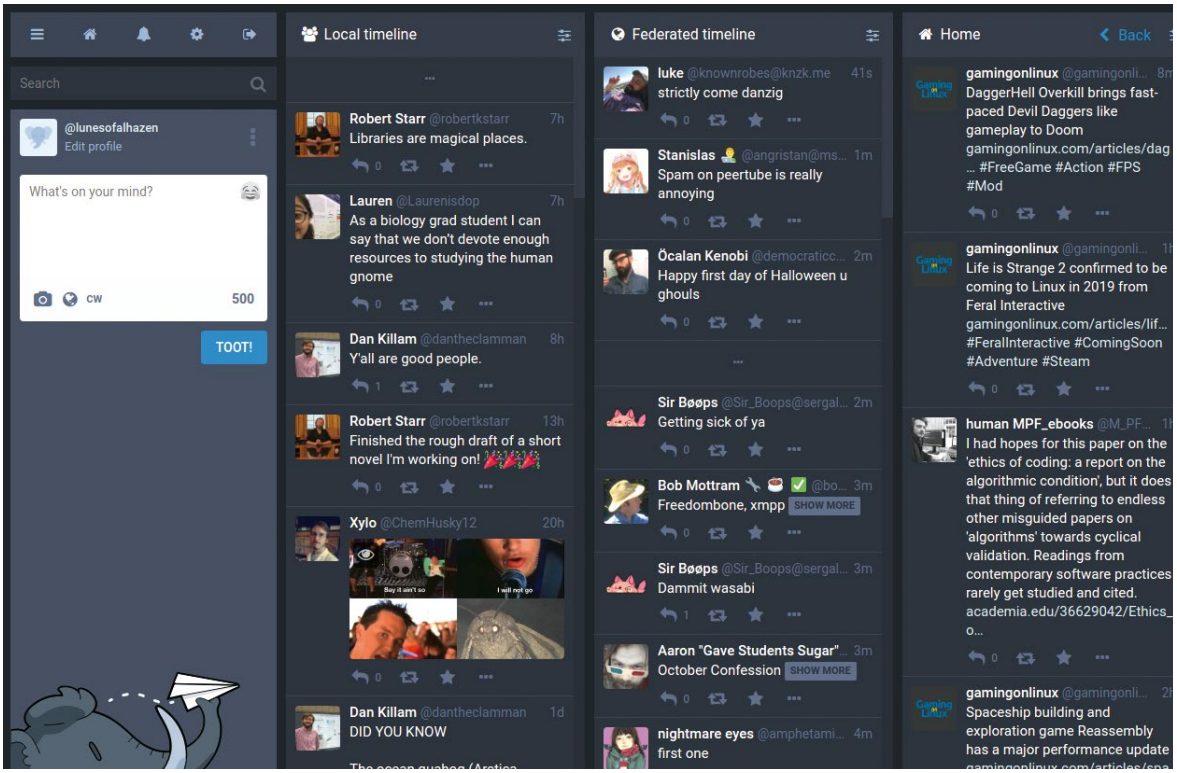
For static images, and those looking for a home for them that isn't Instagram or Flickr, there's PixelFed (**https://pixelfed.org**). Like Mastodon you can host your own Pixelfed instance and federate with other instances, such as the one you'll find at **https://pixelfed.social**.

For a more Facebook, less Twitter, type of decentralised social network, diaspora* (**https://diasporafoundation.org**) is where you should turn your attention. Instances are referred to as "pods" and pods may be based on geographical location, interest, the foresaid funky domain names, or popularity. For the indecisive, The Federation (**https://the-federation.info**) collects opt-in statistics for not just diaspora* pods, but instances for all the networks mentioned here and many more. Federation blurs traditional boundaries, so many of these networks can talk to each other, either through crossposting tools or natively via ActivityPub.

We need to restart the shell to enact these environment changes, then install `ruby-build` and install the required Ruby version:

```
$ exec bash
$ git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
$ rbenv install 2.5.1
$ rbenv global 2.5.1
```

Now we can clone the Mastodon repository, checkout the latest stable release (v2.5.0 at the time of writing), and install the needed Ruby gems and **node.js** packages. Change the `-j` argument in the



Mastodon's layout resembles that of Tweetdeck or Hootsuite. You can find Jonni at @lunesofalhazen@scicomm.xyz, but to be honest he doesn't say much.

`bundle` command to the number of cores on the machine, as follows:

```
$ cd ~
$ git clone https://github.com/tootsuite/mastodon.git live
$ cd live
$ git checkout $(git tag -l | grep -v 'rc[0-9]*$' | sort -V | tail -n 1)
$ gem install bundler
$ bundle install -j2 --deployment --without development test
$ yarn install --pure-lockfile
```

Now we can log out the `mastodon` user and continue with setting up Postgres and the Nginx web server. First we start the database client and create a database user:

```
$ sudo -u postgres psql
postgres=# CREATE USER mastodon CREATEDB;
postgres=# \q
```

Setting up a webserver and https certificates is beyond the scope of this feature, so we'll assume you've got Nginx up and running, as well as a domain name and SSL certificates for that domain. Create an empty configuration for your Mastodon site, say **/etc/nginx/sites-available/lxfmastodon.net.conf** (change this to match your domain). Then copy, paste and change the example.com URLs from the example Nginx configuration at **https://github.com/tootsuite/documentation/blob/master/Running-Mastodon/Production-guide.md** (about halfway down the page). You'll also find instructions for generating and renewing free SSL certificates with Let's Encrypt there. Now we can activate our Mastodon site with

```
$ cd /etc/nginx/sites-enabled
$ sudo ln -s ../sites-available/lxfmastodon.net.conf
```

All the pieces are now in place and we can run the Mastodon setup wizard, which we'll do as the `mastodon` user set up earlier:

```
$ sudo su - mastodon
$ cd ~/live
$ RAILS_ENV=production bundle exec rake mastodon:setup
```

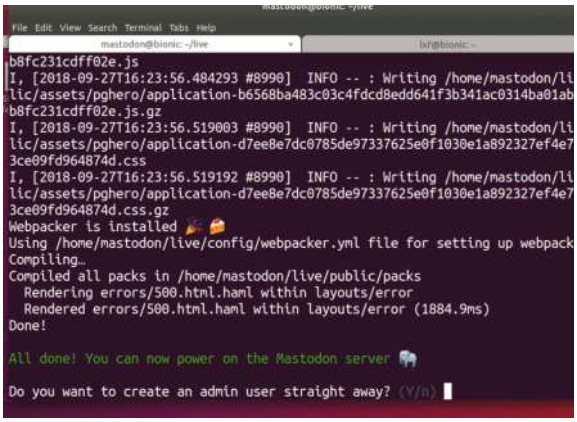
You'll be prompted to enter your domain name, whether to use single user mode (useful for testing purposes), whether you're using Docker (not this time), PostgreSQL details (the defaults are correct unless you changed something – the database user deliberately has no password). Mastodon can send emails (for account verification and recovery purposes) either using a local server or through an external service such as Mailgun. It's possible to run an instance that doesn't send emails too. You'll be prompted to set up an admin user, which you should because you'll want to administer your instance. A password will be generated for you: note it down and change on login.

Set up services

Our work is tantalisingly close to being done, we just need to set up three Systemd services: `mastodon-web`, `mastodon-sidekiq` and `mastodon-streaming` which should be placed in **/etc/systemd/system/**. Again, you can find, copy and paste their contents from the setup guide linked earlier. We can start and enable this trio of services in one shot with:

```
$ sudo systemctl enable --now /etc/systemd/system/mastodon-*.service
```

Now, fingers crossed, you should be able to access your Mastodon instance through your web browser. Log in with the email address you signed password provided, and tell all your friends to sign up.



The final stages of Mastodon's configuration wizard use a lot of RAM and spit out a lot of data, punctuated by reassuring emoji.

THE FATE OF YOUR DATA

"Ever since Yahoo!'s acquisition of Flickr and YouTube's assimilation by Google, people have raised concerns about how their pictures and video are used"

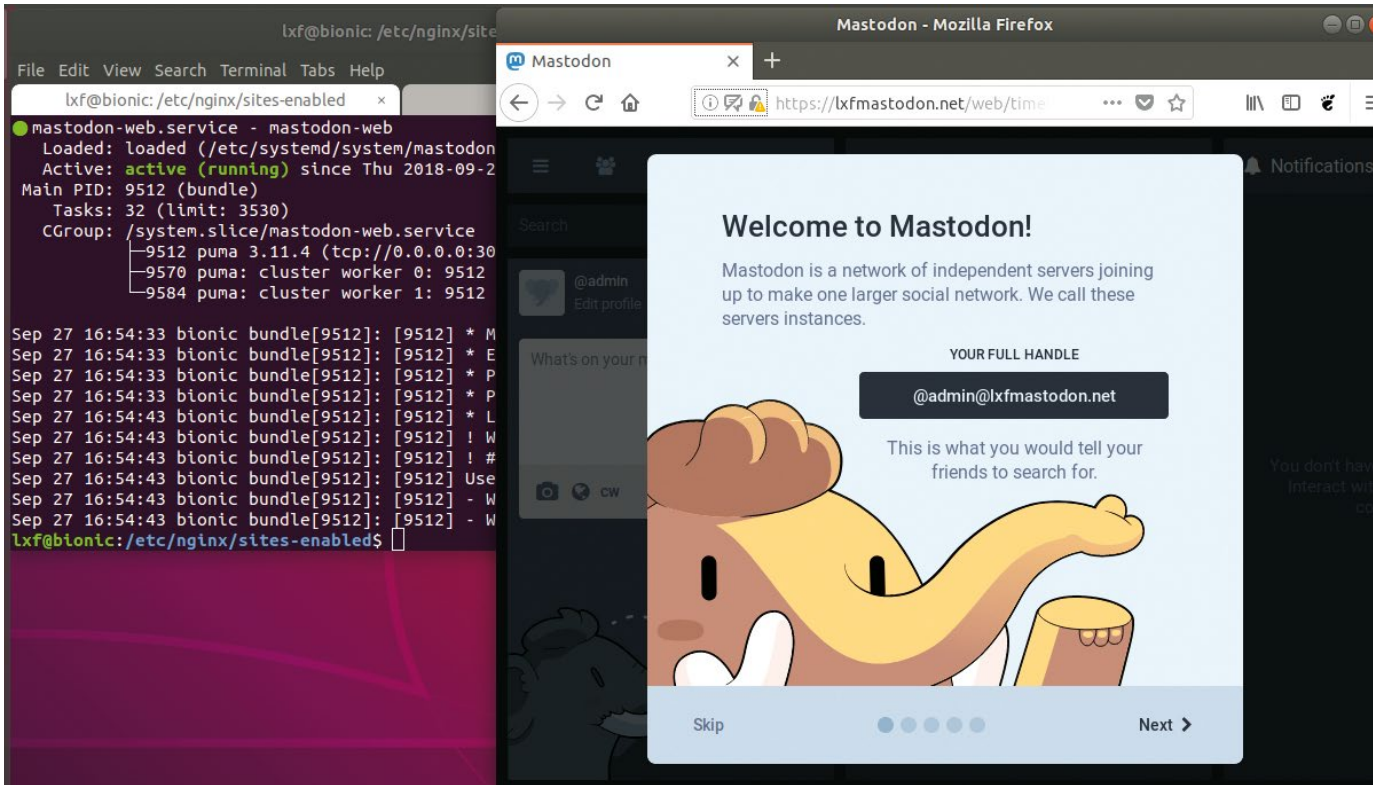
» SOLID AND THE DECENTRALISED WEB

Professor Tim Berners-Lee invented the World Wide Web, so when he announces that his invention "has evolved into an engine of inequity and division" (see **https://medium.com/@timberners_lee/one-small-step-for-the-web-87f92217d085**), people tend to listen. The professor and his team at MIT have come up with a plan to re-invent it.

Built on top of the existing web, the Solid platform hopes to give users the ability to choose where their data is stored, who can see it and how it is used. Solid is a portmanteau of "Social Linked Data", and the idea of linked data is nothing new – Berners-Lee introduced it in 2006, alongside the idea of the semantic web. It enables published data to be contextually linked and sourced from external sources, so that it's no longer beholden to a particular provider or application. If a new, better app comes along, then it can access and re-use the same data as its inferior predecessor.

It's an ambitious effort, but Solid isn't the only projects aiming for a decentralised re-imagining of the web. ZeroNet, the SAFE (Secure Access For Everyone) Network and PiperNet all hope to achieve this in their own ways.

Why try and re-invent centralised web services when you can re-invent the whole entire web?



One final note in case your following becomes popular. Mastodon caches data from other instances, where your users' followers are, and this cache can grow rapidly. Avoid any problems by adding the following to the `mastodon` user's **crontab**:

```
RAILS_ENV=production
@daily cd /home/mastodon/live && /home/mastodon/.rbenv/shims/bundle exec rake mastodon:media:remove_remote
```

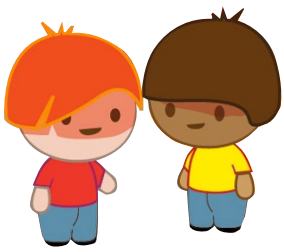
Ever since Yahoo!'s acquisition of Flickr in 2005 and YouTube's assimilation into the Google empire in 2006, smart people have raised concerns about how their pictures and moving pictures are used. In our exclusive **LXF222** interview Cory Doctorow talked about his horror at the new terms that were foisted upon users following Flickr's Yahoo!-ification. Users were forced to associate their Flickr account with a Yahoo! one, or they'd be unable to access their images. Furthermore, content becoming subject to US Federal Law, rather than Canadian Law, was a major concern. Add to that Yahoo! managed to divulge personal data of some three billion accounts (basically anyone that ever had an @yahoo.* email address) in 2013. Yahoo! was acquired by Verizon which subsequently sold off Flickr, not to mention the majority of their other assets. But the concern endures.

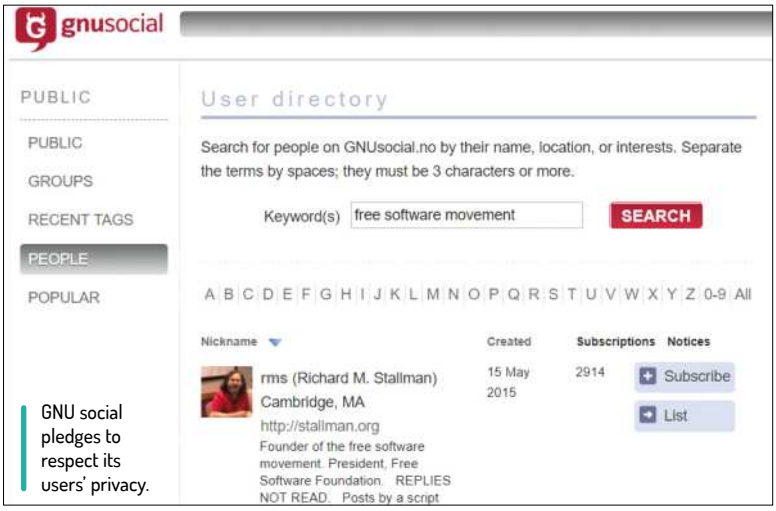
And another thing...

Now let's turn to Google, and in particular its unified privacy policy edict of 2012. Ostensibly it's made its services (at the time there were about 60, and most people used at least one) simpler to use, but it also afforded them a tremendous amount of data insight. Before, data relating to each service was siloed, so while it was still collecting huge amounts of data from scanning emails (it no longer does this), keeping track of searches, and running all kinds of nebulous algorithms on YouTube uploads, there was no attempt to correlate across these services. After the fact, users had a choice: either let Google associate all this data with a single identity, or stop using Google's services altogether. Today, with about two billion people using Android phones all associated with a Google account and all merrily transmitting location data, the problem is stark (*that's why I use LineageOS – Ed*).

It works! Although we cheated and used a self-signed certificate and a bogus domain name, but you don't need to know that...

It's easy to target mega corporations, but it's the whole centralised model that's at fault. A pioneering company with all good intentions (which is exactly what Flickr was) could at any point find itself swallowed up by some awful monster. Data is then effectively held hostage until users bend the knee to whatever unsavoury policies said monster demands. In the decentralised model, if one instance turns rogue, or is taken over by a rogue, then that rogue instance can be blacklisted. If a whole project turns to the dark side, then it would be trivial to fork the software and rebuild the network. LXF





GNU social pledges to respect its users' privacy.

The pixel-doubling option is handy if you don't like squinting at tiny explosive barrels and other sprites in OpenRA.

OPEN SOURCE GAME ENGINES

Like the Revenant in Doom II, **Jonni Bidwell** brings the corpses of lost video games back to life with a lil' bit of FOSS magic

None one of us at *Linux Format* are as young as we used to be. But our considerable collective age means we remember some fantastic games.

Some of these go way back: *Exile*, *Elite*, *Repton*, *The Last Ninja* on the BBC model B, *Uridium* and *Maniac Mansion* on the C64. Most of us spent most of our youth playing Amiga games, *Monkey Island* taught us how to swordfight, and who could forget those Bitmap Brothers classics: *Xenon 2*, *Speedball 2* and *The Chaos Engine*? Just talking about them makes us want to fire up an emulator and relive those halcyon days.

But emulation can be clunky, and getting hold of ROMs and the like is tricky. Even playing early PC games through DOSBox can be troublesome. Thanks to the magic of open source though, we can do better. Many classic titles have had their game engines reverse engineered or revamped, so that the original game assets can be used seamlessly on a modern system. This can make for a completely authentic experience, or allow new features (better graphics, proper network play) to be added.

More importantly, it enables us oldies to show those young whippersnappers what gaming was like back in the day. Before the days of running around in a slick 3D environment, chasing in-game purchases, and learning to swear in Russian by listening to in-game chatter.

The LucasArts adventure games have achieved legendary status. Its first graphical effort, 1987's *Maniac Mansion*, set a new standard for the genre. The game featured a bespoke scripting engine, imaginatively titled Script Creation Utility for Maniac Mansion (SCUMM), which provided high-level routines for common game primitives: characters, locations, dialogues and inventory. This not only made coders' lives easier, since they could work with human-readable scripting commands rather than assembly, but also made the task of porting the games much easier, since this only required porting the engine to the new platform. The SCUMM scripts and game assets could more or less be used as is.

SCUMM came about because many of LucasArts' programmers were originally mainframe[1] programmers, and found it much easier (and faster) to compile code on those machines before porting it to the native platform. By writing clean and portable code for SCUMM, coders Ron Gilbert and Chip Morningstar afforded themselves the luxury of being able to write in a flexible scripting language that could be rapidly compiled for multiple platforms: the C64, Amiga, Atari, PC and Mac. SCUMM also made possible all kinds of things never before seen in adventure games: point 'n' click gameplay, multiple characters and background tasks/animations. It was arguably the first game engine, in the sense that it decoupled game assets and gameplay. Amazingly, with only modest modernisations, SCUMM would go on to power a decade of adventure games. These included the *Monkey Island* series, *Loom*, *Day of the Tentacle*, *Full Throttle*, *The Dig* and more.

Now, picture yourself back in the early 2000s, and suddenly you want to relive those memories of playing *Monkey Island 2: LeChuck's Revenge*, but without the wrist ache that came with swapping the 11 disks that housed the Amiga version. Well, that game had a DOS release so you could probably get it working in Windows XP, but no, even back then you were pretty militant about using Linux, so that won't fly. *DOSBox* had just been released, but so had an ambitious piece of free software called *ScummVM*, which reimplemented SCUMM natively, not just for one game, but for a whole bunch of them. Even in its infancy, this amazing software caused a stir amongst Linux gamers (all three of them), since it provided another avenue for Windows (and Mac) titles to be played on Linux.

ScummVM came about as a result of one Ludvig Strigeus' desire to better understand adventure game engines, so that he could write his own. He started by reverse engineering *Monkey Island 2* and eventually

Once you've got the game's file you're exactly one click away from a quick Skirmish in Command and Conquer.

» THE SMALL MATTER OF LEGALITIES

Dabbling with emulation and the like can fairly rapidly lead to legally murky waters. In the US copyrights last for 75 years, so even ROM files of early arcade games are off-limits. Even if you own an original copy of the game on one platform, you're not entitled to download another (getting data off Amiga formatted disks requires special hardware, for example) for use on an emulator. Just because a title is no longer available doesn't entitle you to download it.

However, old PC games are readily available for cheap on auction sites, fairs and stoop sales (which is like a yard sale if you live in Brooklyn). The century old "first-sale doctrine" covers this legally. For older titles you'll need a USB floppy drive, and apparently computers now don't come with optical drives either so you might need one of them, too. For the reinvented game engines under consideration in this feature, we're only interested in getting the asset files from the original media. This circumvents any DRM issues we may run into if we were running the original binaries.

OpenSC2K was a remake of the classic (and dromedary-joke-packed) *Sim City 2000*. Alas, in July GitHub received a DMCA takedown notice that obligated it to take the *OpenSC2K* repository offline. That repo included assets from original game, and since *Sim City 2000* is still sold by EA for about £5 (though the company has offered it for free in the past) the project soon attracted the attention of EA's legal team.

came up with an interpreter capable of playing the game. This would be the first version of *ScummVM*.

Meanwhile Vincent Hamm was independently tackling SCUMM from a different angle, by investigating the scant documentation available online and investigating the scripts inside *Maniac Mansion* and *Zak McKracken and the Alien Mindbenders*. The two combined their efforts and worked on *ScummVM*, somewhat haphazardly at first. However, interest in the project skyrocketed when it was featured on Slashdot, a popular thing of the time called a website, and a slew of developers wanted to join the party.

LUCASARTS DONE GOOD

"SCUMM was arguably the first game engine, in the sense that it decoupled game assets and gameplay"

Initially, *ScummVM* was written in C and supported a handful of games: *Monkey Island 2* (how the project began), *Zak McKracken and the Alien Mindbenders* and *Indiana Jones and the Last Crusade*. The master source tree was stored on Strigeus's machine and Hamm would contribute there. By 2002 the project had been rewritten in C++, which brought portability, and supported tens of SCUMM games, and even a non-SCUMM adventure: *Simon the Sorcerer*. This led to some naming controversy, but in the end the original name stuck.

Today, *ScummVM* supports hundreds of games and tens of engines: Sierra's AGI and SCI interpreters, Coktel's *Gobliiins* (sic) series, and Revolution Software's Amiga classic *Beneath a Steel Sky* (which is available for

1) www.gamasutra.com/view/feature/196009/the_scumm_diary_stories_behind_.php

free on **gog.com)**. New features (more bug fixes) are being added all the time, so check out the project's GitHub at **https://github.com/scummvm/scummvm**.

In 2002 LucasArts sent a cease and desist letter to the *ScummVM* team, but in the end (the process took about four years) the two parties came to an agreement whereby the project could continue. After this, engine reimplementation began to be recognised as a non-infringing activity, and further discussions with rights holders were largely positive. Revolution Software and Adventure Soft even provided source code to help support the engines used by their games. Two-thirds of the 25-year anniversary edition of the *Myst* trilogy are powered by ScummVM (*Myst III* is powered by ResidualVM, which reimplements its Sprint engine, as well as GrimE, the successor to SCUMM used in *The Curse of Monkey Island* and *Grim Fandango*). You can read more about ScummVM at **www.pcgamer.com/how-scummvm-is-keeping-adventure-games-alive-one-old-game-at-a-time**.

As you might imagine, it's generally illegal to redistribute assets from commercial titles. The box on the previous page goes into more detail, but suffice to say you'll want to hold off downloading anything from random websites. Back in July 2018 popular emulation sites LoveROMS and LoveRETRO found themselves on the receiving end of what's likely to be a very expensive lawsuit from Nintendo. Prompted by these legal actions, the popular EmuParadise (**www.emuparadise.me**) removed all download links from its site.

Many games are referred to as 'abandonware', which refers to them no longer being sold or supported by their original manufacturers. This term doesn't offer any legal protections, though: copyright holders may be pragmatic and choose to not worry about infringement, they may even offer titles for free if enough people ask nicely. Or they can be like Nintendo and have a whole page dedicated to how hard it will come down on you if you don't respect its property (**www.nintendo.com/corp/legal.jsp**).

The spice must flow

For *OpenRA*, the re-implimentation of the classic *Command and Conquer* (CnC) engine, the situation is more mellifluous. To celebrate the 12th anniversary of the original *Command and Conquer* (known in some circles as *Tiberian Dawn*) in 2007 EA released the original ISOs as freeware. To celebrate its next anniversary (and also the release of *Red Alert 3*) in 2008 EA released the original *Red Alert* and *Tiberian Sun* ISOs as freeware. It no longer hosts these, but has stated that it's happy for others to redistribute them. As a result, *OpenRA* can find and download all the required files (sans music and video cut scenes) to play these three titles if you don't have the original install media.

The 1995 Discworld game (featuring the voice of Eric Idle) can be brought back to life with ScummVM.

In *OpenRA* parlance, the games the engine brings back to life are known as mods. The *OpenRA* project has embraced new packaging formats so you can download AppImages of the official mods straight from **www.openra.net**. These downloads come in two flavours: stable releases and playtests. The latter are previews featuring the latest engine developments, and so may not be as stable as the former.

Besides the official mods, a number of community offerings are available (see **www.moddb.com/games/openra/mods**). Work is underway to decouple the *OpenRA* engine from the 'official' *CnC* mods that it powers, so that it can be used as a general-purpose real-time strategy engine. The project already provides utilities such as a map editor and a Source Development Kit (SDK) to help budding developers build the games of their war-like dreams. Readers may also remember the *Krush Kill 'n' Destroy* games of the late 90s. Those readers may be thrilled to hear that work is underway to revive those post-nuclear war titles using the *OpenRA* engine.

Another related project is *CnCNet*, which has been around since 2009. They provide launchers that enable the CnC series (including all expansions and mods) to be played exactly as their fans remember, with the added bonus of internet play. Recently, it reimplemented *CnC's* DirectDraw renderer, in GDI (for newer versions of Windows) and OpenGL (for Wine players). *CnCNet* supports the titles released for free, as well as *Tiberian Sun* (*OpenRA* is working on this one), *Red Alert 2* and *Yuri's Revenge*. Linux packages (for Fedora, Debian, Ubuntu and even Arch and derivatives) are available for the free *CnC* titles. These are just scripts that set up an appropriate *Wine* prefix and download the game files and the *CnCNet* launcher.

The other titles supported by *CnCNet* are only available through EA's Origin platform as part of the *Command & Conquer Ultimate Collection* (priced at £25), but getting *Wine* and Origin to play nice together can be tricky. You may be better off seeking out optical media for these, which the *CnCNet* launcher is more than happy to work with.

The original Westwood Online servers were transitioned to the community-run, EA-sponsored, XWIS server in 2005, which is still alive today. The newer *CnC* titles ran on GameSpy servers that were shut down in 2014. This shutdown affected a huge number of titles, many of which moved to other networks. The *CnC* titles moved to the C&C:Online (**https://cnc-online.net**), where they are alive and well today.

Till it be Morrow(ind)

Morrowind is the third installment in Bethesda's highly acclaimed *Elder Scrolls* RPG series, launched in 2002. Its vast, open world was powered by the Gamebryo engine, which has been re-implemented by *OpenMW*.

The haunting Bitter Coast as rendered by OpenMW.

Credit: CCBY 3.0 Laggahan, wiki.openmw.org

Besides addressing long-standing bugs (*Morrowind* received its last official patch in 2003), the engine adds new features and, of course, enables the game to be played natively in Linux. The original game assets are required, but that's fine because you can get hold of a DRM-free copy of *Morrowind* (the full-priced Game of the Year edition) from **gog.com** for £13. It's often on special offer too so add it to your wishlists if you're short of coin.

OpenMW modernises the original game by being able to render distant terrain (increasing the draw distance on the original reduced performance horribly), use TrueType fonts and works with all the multimedia formats supported by FFMPEG. Some third-party mods for the original *Morrowind* will work with *OpenMW*, but this isn't a strict goal of the project at present. Many mods relied on bugs in the original engine or assumptions about undocumented quirks.

OpenMW also comes with *OpenMW-CS*, which makes it possible for users to create their own assets: characters, maps, quests... the whole shebang. This enables ambitious users to form entire games based on *OpenMW*. Skilled 3D sculptors can use *Blender* to create models of dragons and dungeon-paraphernalia, and use export them using the NIF (a proprietary format used by many other games) or OSG (*OpenMW*'s own open format) plug-ins.

There's an inordinate number of other open source game engines which we haven't had a chance to discuss (*maybe we would have if someone didn't spend all their time playing Red Alert instead of writing this feature – Ed*), Check out the impressive list at **https://osgameclones.com**. But in the name of nostalgia we'd like to give a nod to Freeserf (**https://github.com/freeserf/freeserf**), which makes it possible to play (again with the original data files) the classic medieval kingdom simulator *The Settlers* from 1993. Readers, do let us know if you find any classics from your youth that have been brought back to life through the magic of open source (*just don't tell Jonni as he's distracted by pixelated things and old jokes – Ed*). LXF

» BUT DOES IT RUN DOOM?

It's become something of a tradition to run shareware classic *Doom* on all sorts of hardware never intended for running Doom (see for example **www.gamesradar.com/12-things-that-prove-that-doom-will-run-on-literally-anything**). However, most of this silliness is enabled through the many reworkings of the *Doom* engine, to which id Software released the source code in 1997 (it's available at **https://github.com/id-Software/DOOM**).

One of the earliest ports was *GLDoom*, which brought OpenGL support to *Doom*, but this was a DOS-only affair. *Boom* overhauled the *Doom* engine, fixing many bugs and oddities, as well as removing limitations that no longer made sense for (then) modern hardware. *Boom* wasn't initially open source, but the code was released in 1999, which gave rise to *LxDoom* for Linux and *PrBoom* for Windows. These two projects eventually merged under the *PrBoom* moniker. *PrBoom* has since been ported to many platforms.

There are many other more *Doom* clones, some of which add new and advanced features, but some people treasure the original. For those people there's *Chocolate Doom (see* **LXF226***)*. It recreates as authentically as possible the DOS experience, even going as far as to re-implement bugs that were later fixed. This means original demos can be played, the imps are as dumb as ever, and slight artefacts that some levels of the original relied on are faithfully reproduced.

It's Chocolate Doom running on a chainsaw. Just don't play while you saw.

Roundup

Flowblade » Kdenlive » LiVES » Openshot » Shotcut

Shashank Sharma
By day Shashank is a New Delhi trial lawyer, but by night he's an open source vigilante!

Video editors

Desperate to turn his amateur attempts at shooting film into praise-worthy masterpieces, **Shashank Sharma** tries his hands at video editors.

HOW WE TESTED...

We're running the latest offering for each of the tools featured in this *Roundup* on top of a quad-core machine with 12GB RAM. Apart from vast amounts of RAM and processing power for optimum performance, you'd also need plenty of spare disk space to horse around with video editors.

All the projects are fairly popular and many distribution feature them in the software repositories. Some of the tools featured here also ship Snap and AppImage packages for easy installation. Apart from ease of installation, documentation is vastly important to help new users acclimatise to the vast feature set contained within each of these projects.

We'll discuss some of the most useful features these tools offer and whether their interface is intuitive or if it gets in the way of you jazzing up your home videos. While cellphone cameras are all the rage, many people still use dedicated devices to shoot film, so we'll also consider the different file formats supported by these tools.

Whether it's still photography or a film recording some minutiae of the daily life for posterity, some people just have a natural knack for framing shots. They can make even a plastic bag blowing in the wind look mesmerising. With just a dash of creativity but far more patience, you too can transform your amateur captures into professional-grade videos, complete with a background score, if you like. But this is only possible if you're willing to put in the time and effort it takes to master these behemoth, feature-rich projects.

We've selected the tools that are easy to use for new users, yet offer enough features to be of interest to skilled video editors. With ever-decreasing prices of consumer hardware, it's no surprise that these resource-intensive tools have continued to find favour with the developers and users over the past 15 years or more. The youngest application in our list, *Flowblade*, was first released in 2011, while the oldest, *Kdenlive* and *LiVES*, have both been around since 2002.

Using these tools, you can add effects to your videos, cut out unnecessary footage or out of focus parts, move stuff around, and otherwise splice or combine different clips into a single meaningful narrative. The world is your editing oyster!

Installation

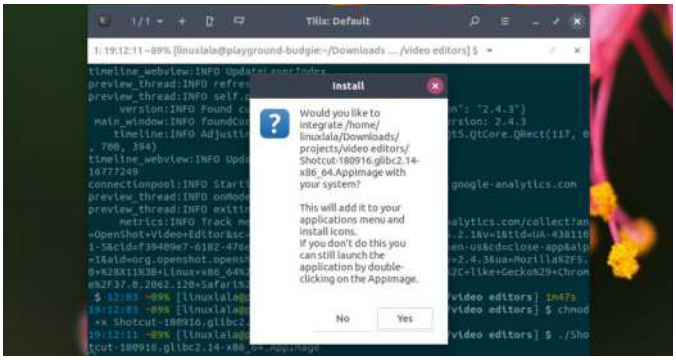
How easy is it to cross the first hurdle (turning the lights on)?

You really learn to appreciate the beauty of the package management tools on Linux distributions when you attempt to install mammoth projects like the ones featured in this *Roundup*. Not only do these tools require a number of different third-party applications and libraries to provide many essential functions, you also need a large number other libraries and resources. Worse still, some of the additional software have vast dependencies of their own.

This is why distribution agnostic packaging formats like AppImage and Snap are finding great favour with many different application developers. Another advantage with such packages is that they require no installation, making the software portable. You merely have to make the downloaded file executable by running the `chmod +x <filename>` command, and you're done.

Kdenlive, *OpenShot* and *Shotcut* each provide portable 64-bit AppImage packages. The *Kdenlive* help manual also describes the different resources from where you can grab deb or RPM binaries, if you prefer applications to be integrated into the desktop.

Apart from an official deb binary package, *Flowblade* doesn't ship any binaries for other popular distributions. The download page on the website recommends using the software repositories of your distribution for installing the project, but also cautions you that the latest release might not always be on offer in the software repositories. Thankfully, you can find RPM packages for the latest release on third-party websites such as RPMFusion.



Kdenlive and Shotcut offer to create entries in the menu when you first run the AppImage binaries.

LiVES doesn't produce binary packages for different distros itself, but recommends Ubuntu users to install it using a PP. Fedora users can similarly find RPM packages for the latest release on RPMFusion. If you're confident in your dependency-resolving skills, you can compile it yourself from source. But for most of the projects featured here, this is ill advised.

When you launch *LiVES* after installing it, unlike the other tools, it starts a configuration wizard that checks if your system has all the dependencies installed. Next, you'll be asked to choose an audio player, whether *pulse audio* (recommended), *jack audio*, *sox* or *mplayer*. Finally, you're asked to choose between two different startup interfaces: clip edit or the multitrack mode.

VERDICT

FLOWBLADE	**5/10**	**OPENSHOT**	**10/10**
KDENLIVE	**10/10**	**SHOTCUT**	**10/10**
LIVES	**7/10**		

While Flowblade only works on Linux, all the others support Windows and Mac.

Documentation

No one can RTFM if there isn't one!

Flowblade features quick instructions on how to use the application, but the lack of suitable screenshots make following the terse instructions difficult. Worse still, the instructions are outdated and don't reflect the changes introduced with recent releases.

Kdenlives fares far better, with a thorough user manual hosted on **https://userbase.kde.org/Kdenlive/Manual**. You'll also find a forum board discussing different aspects of the project such as Installation, Video Effects & Transitions and Audio. Users can also share their own content in the *Kdenlive* Gallery forum.

First published in 2009, the *LiVES* user manual isn't all that relevant any more, and is of no assistance to novices. The Beginner's Quick Introduction is a better prospect, coupled with the several video tutorials available on the Documentation page of the website. You can also ask for help in the lives-users mailing list.

Every section of the *Openshot* use guide is replete with helpful screenshots of the tool. If you're already familiar with video-editing software, or even if you only have a general awareness of the process, the Quick Tutorial runs you through the broad steps. You can then navigate the sidebar to delve deeper into the usage. The forum boards hosted on **openshotusers.com** aren't active and there hasn't been any new posts in almost a year. But the dedicated sub-forum on user-recommended tutorials are definitely worth some of your time.

Complex operations such as using the multi-track editing mode on LiVES require patience, and spending time with the documentation.

Shotcut features a Getting Started guide in several languages, including English. Also on offer are video tutorials covering key topics such as Multitrack Timeline Basics.

VERDICT

FLOWBLADE	**7/10**	**OPENSHOT**	**10/10**
KDENLIVE	**10/10**	**SHOTCUT**	**8/10**
LIVES	**8/10**		

Flowblade needs to rethink its approach to documentation.

User experience

Is the interface intuitive to a mere human or only to a Cyberdyne T-800?

Whatever their interface and design philosophy, all video editors perform similar function. To begin, the tools enable you to manipulate a single video file, or a series of them, and arrange them into a sequence. All along, you can deploy different effects to smoothly transition from one clip to the next. You can also insert a title to the video, to give it proper introduction.

In common parlance, a track comprises video clips, effects and title file. The tools make it possible to work in parallel on several different tracks, where one is for audio, another for effects, and so on. The tracks are all visible on a single interface referred to as the timeline. The final part of working with these editors is rendering, which turns your work into a chosen output format, such as DVD, online publishing, and so on.

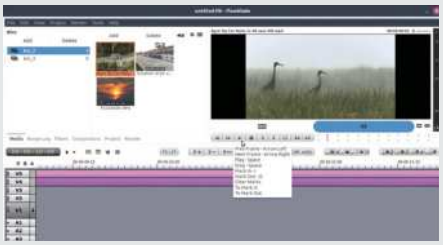
We're looking for a tool that seamlessly integrates these different tasks into an easily navigable interface.

Flowblade 6/10

Flowblade presents a simple interface when you first launch it. While the interface is riddled with buttons, you don't feel overwhelmed with options. You can't maximise the application window, and it appears as if the interface cuts off abruptly at the bottom, but don't worry, this is all there is to the application window. The timeline at the bottom comprises a series of grey rectangles, on to which you will drag-drop the clips that make your film.

All your media files are stored in what *Flowblade* refers to as bins and your project must have at least one bin. To apply a filter, you must drag-drop it on to a clip on the timeline. The application automatically pops open a dialog providing useful information on what the filter does.

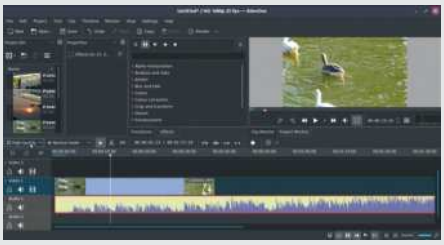
The limited number of undo/redo actions and insufficient documentation are key reasons why we can't recommend *Flowblade*, despite its clean interface.



Kdenlive 9/10

Before anything else, you must go through the *Kdenlive* documentation,. Not only will this serve to help you make sense of how *Kdenlive* works, it's also an excellent introduction to video editing in general. Of course, the different tools have different approaches to the workflow, but understanding the process will help you easily master the different tools.

The left-most panel holds the different clips that you wish to work with. The right of the screen is reserved for monitors. *Kdenlive* features a Clip Monitor, which displays the clips you're working with. Click the Project Monitor button to view how the finished project will look like with all the effects and transitions. Below the monitors are the timelines, where you will work to manipulate the audio and video files. If you drop a video file on to an audio track on the timeline, *Kdenlive* retrieves the audio from the clip to work with.



Supported file formats

Even cinematic masterpieces are encoded MP4, right?

At the very least, the video editor should be able to work with a variety of file formats and produce output files that are fit for appearing on different platforms, such as online publishing.

Thanks to their reliance on *FFmpeg*, most of the tools on our list can work with almost any video format, and render files in your desired format. All the tools present a large scrollable list of formats you can export files your project to, after you're done editing the videos.

With *Flowblade* and others, you can also create a profile of the project, which depends on the source video files you wish to work on. All the tools are capable of correctly identifying the source of the files, and will suggest changing the profile if it's different from the video files.

LiVES is different from the others in that it relies on *mplayer* decoder when working with the different formats. Both *Kdenlive* and *LiVES* support Firewire input, but this option isn't yet available for the other tools.

Another useful feature of LiVES is that it can be used to capture external windows via mouse clicks. Coupled with its support to record audio from an external source, you can easily use the application to create professional-grade screencasts.

You can similarly use *Openshot*, which is quite handy at capturing feed from a webcam. Apart from choosing the video codec, resolution, aspect ration and so on, the tools also choose the audio channels, bitrate and more. The default setting for each of these varies according to the chosen output format, which can be Webm, Matroska, OGG, SWF and more.

VERDICT

FLOWBLADE	**8/10**	**OPENSHOT**	**8/10**
KDENLIVE	**9/10**	**SHOTCUT**	**8/10**
LIVES	**9/10**		

Shotcut and Openshot plan on introducing Firewire support at some point down the road.

LiVES 6/10

Unlike the other applications, which heavily rely on buttons within the interface to handle the different operations, *LiVES* is old school. It features a plain old toolbar at the top with different heads such as File, Edit, Play, Effects and Audio. along with a handful of buttons.

The tool defaults to the Clip Editor mode when you run it. You begin by adding files from the File menu. For each file you add to your project, *LiVES* will display the first and last frame in the video clip and the total number of frames. If the video has sound, it's relegated to the audio track at the bottom of the window.

To edit clips, you must launch the Play window, which has its own playback control buttons. Unfortunately, these buttons disappear as soon as you hit Play, making it difficult to stop the video, or select the portion of it you want to work with.

Openshot 8/10

Unlike *LiVES*, where a lot of the toolbar entries remain disabled until after you've added files into it, you can freely navigate *Openshot* to get a sense of how it organises the workflow without importing any files, or starting a new project.

As with all the other tools, the timeline is restricted to the bottom of the interface while project files, available effects and transitions can be accessed by pressing the appropriate buttons on the left panel. You must resize the panel at the bottom to be able to access the complete timelines. When you overlap two clips on the timeline, *Openshot* automatically creates a smooth fade between them. With the other tools, you must work to put in a fade effect.

Whereas the other tools feature separate tracks for video and audio clips, *Openshot* enables you to add all files on to a single track, which makes things more convenient for new users.

Shotcut 7/10

Shotcut and *Kdenlive* have the distinction of having cluttered interfaces. While the latter makes up for its with its vast documentation, the reliance on video tutorials exclusively to introduce *Shotcut* feels ill-advised.

The interface features a toolbar at the top with several useful buttons. Clicking these opens the relevant element within the interface itself. It's easy to miss the new elements on the interface, since everything's fairly crowded already. For instance, keep your eye at the bottom of the window as you alternatively click Timeline and Timeframe. You'll notice the different elements appear without any fanfare. Ditto when you switch from Filters to Export, which affects the left panel.

For each new element that pops up within the interface, you're also presented with a number of additional buttons to control the different settings

System requirements

Nobody enjoys a slow-motion car chase through rush-hour traffic.

Video editing is resource-intensive work. As you grow accustomed to these tools and begin to work with it to process your home videos, you're going to want to put their batch processing abilities to use, to save time. Although you don't need oodles of disk space or RAM, you can't get very far with these tools on low-spec machines. While you don't need a dedicated high-end rig to get started with these tools, you can't expect to go very far with a repurposed netbook, brought back to life with a lightweight Linux distribution.

Kdenlive and *Flowblade* don't provide minimum system requirements on their respective websites. This is perhaps because of their proxy editing feature. When enabled, proxy editing replaces your original clips with lower-resolution alternative. Because these low-spec proxy clips hardly require any computer power, you can carry out all the operations and enjoy fluent playback.

This feature isn't available in the other applications, and so *Openshot* and *Shotcut* both recommend at least 4GB of RAM to be installed in your machine. For optimum performance, a 64-bit multi-core machine with at least 2GHz process is ideal. Although *LiVES* only recommends 512MB RAM, it was the slowest of all the tools to launch on our quad-core 2GHz test machine which was fitted with 12GB RAM.

Because the requirements depend on the source video files that you work with, *Shotcut* provides a guide on choosing hardware. It recommends at least 4GB for SD, 8GB for HD, and 16GB of RAM for 4K videos. You're also advised to get at least one 2GHz core for SD, two cores for HD, and four cores for 4K videos.

VERDICT

FLOWBLADE	**9/10**	**OPENSHOT**	**8/10**
KDENLIVE	**9/10**	**SHOTCUT**	**8/10**
LIVES	**8/10**		

You can't go wrong with providing as much RAM and CPU cores as possible for these video-editing tools.

Editing aids

Effects, transitions and music.

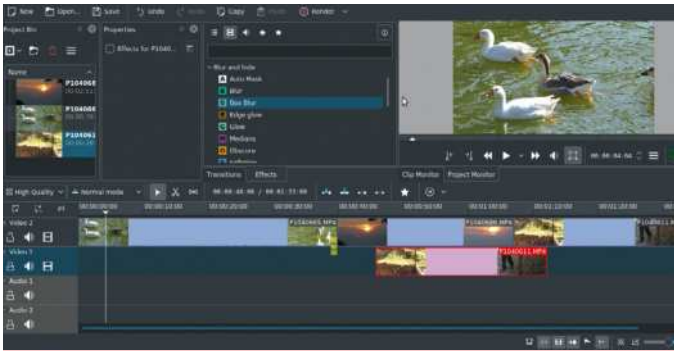
The video editors featured in this *Roundup* can help you weave a compelling narrative by manipulating the different clips and using various effects to enhance their appearance. At the minimum, you need to be able to make smooth transition from one clip to the next.

Although rather equally matched, as you spend more time with them, you'll come to realise that some tools are more intuitive at manipulating files than others.

Shotcut, for instance, refers to these as Effects, and clicking the Effects button at the top presents a panel on the left from where you can introduce different video, audio and other effects into your clips. The application resorts to use of buttons to segregate the different effects, but unfortunately there aren't any helpful tooltips to assist users.

Kdenlive comes across as far more polished on this front, offering a text list of all available effect, split into different categories such as Artistic, Audio, Blur and hide, which makes for easy selection. It also enables you to filter the available choices into video and audio.

If a clip has 540 frames as per *LiVES*, when deploying an effect, it pops open a dialog advising you that it's adding an effect onto all 540 individual frames. The application is unusable during this process, which is far slower, compared to all the other tools.



Be prepared to constantly resize the different elements in the application interface, otherwise you'll only ever get to see a small part of them.

In stark contrast with the other applications, *Openshot* presents helpful pictures to depict the different effects and transitions such as cross, distortion, hue saturation, wave and so on. This is incredibly helpful for novice users, because the alternative is to deploy an effect or transition to determine what each does, as you must with the other tools.

VERDICT

FLOWBLADE	7/10	OPENSHOT	8/10
KDENLIVE	9/10	SHOTCUT	8/10
LIVES	7/10		

The scores here are more reflective of how easy it is to deploy the different effects within each program.

Extra features

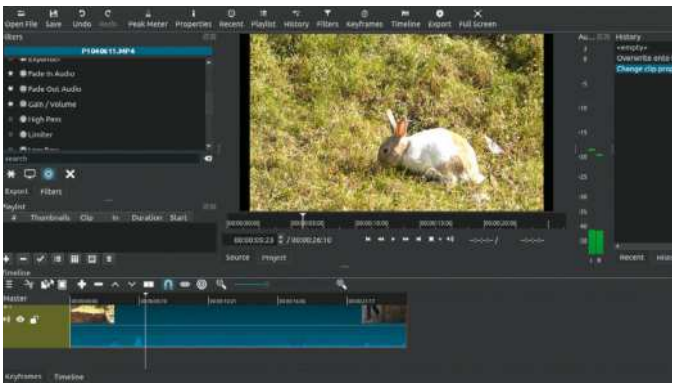
What makes them special?

One of the most unique features of *Flowblade* and *Kdenlive* is the ability to use proxy editing. When original media files are far too demanding of resources such as CPU or disk bandwidth to allow for responsive editing, the tool makes it possible for you switch to working on proxy clips instead, saving system resources.

Flowblade also enables you perform batch rendering operations and split audio from video files. This option leaves you free to change the soundtrack and you can also insert watermarks on to your files.

This is where *Kdenlive's* custom layout feature comes in. Users can tweak the layout to their liking, depending on their workflow. The tool makes it possible to save your custom layout so you don't have to make the necessary changes every time you launch the tool. Best of all, it also enables you to configure the keyboard shortcuts to match your custom layout. Because video editing is a creative process requiring frequent changes, the automatic backup of project files is incredibly handy.

If you want to introduce additional functionality to *LiVES*, you can do so by writing your own custom plugins, which can be in Perl, C or C++, Python and so on. It automatically scales when necessary, depending on the detected hardware. This is why the tool has been successfully on the most number of devices, including x86, amd64, PPC and xBox/x86. Like *Kdenlive*, *LiVES* can also preserve backups of your project and should the application ever crash, you can attempt to recover the files you were working on by starting it with the `lives -recover` command. You can also use the application to load single images or directory of image files in numerical order and assemble them into videos or slideshows.



The right sidebar on Openshot features a History record of all your actions, starting from the time you start a new project.

Openshot provides real-time previews so you can quickly assess your work. It can also be used to create 2D animation and adding 3D animated titled and effects to your clips.

Shotcut offers unlimited undo and redo for playlist edits including a history view. You can also export single frame as image or video as image sequence.

VERDICT

FLOWBLADE	10/10	OPENSHOT	8/10
KDENLIVE	10/10	SHOTCUT	8/10
LIVES	10/10		

Shotcut and Openshot are versatile, but lack proxy editing and custom plugins.

Video editors
The Verdict

With the exception of accidentally overwriting or deleting a file, there's precious little that can go wrong when you test drive a new software, right? While this may be true for most benign software, such as text editors and web browsers, it certainly isn't the case with video editors of the like that we featured in this *Roundup*.

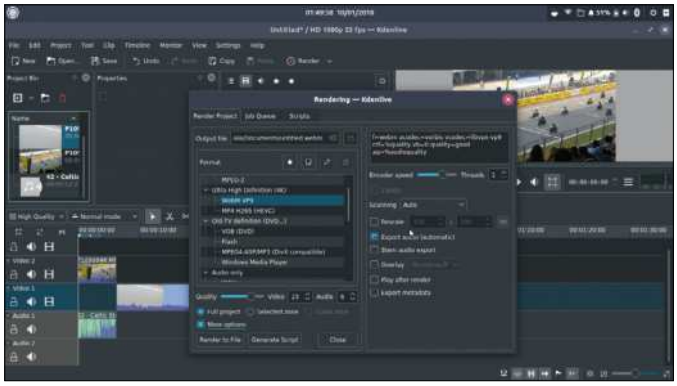
All the tools feature a myriad combination of menu bars, tabs, sidebars and buttons. It'd be unfair to these tools for us to declare any one style or appearance as superior to the others, because each of these tools boast of vast user communities, which suggests that there are ample number of users that favour their user interface.

More than anything else, your choice of application will ultimately come down to which one offers intuitive workflow, because they're all well matched in terms of what you can achieve with them. Furthermore, your choice is something that will vary for every individual. If you're anything like most other people, you'll begin to work with one tool, and over time as you become immersed into the art of film editing, learn to appreciate the different offerings of each tool. It wouldn't surprise us if you even choose to work with multiple tools when working on a project. So, for example, you might use one for their transitions and effects, and another for manipulation audio.

If you're a novice just getting started with video editing, you might find feel daunted by the sheer scope of the possibilities. This is why documentation is so important, and the biggest reason for deciding on *Kdenlive* as the winner.

Flowblade's documentation wildly conflicts with its current design, which makes it unlikely new users will get very far with the tool. Although there's little wrong with the project itself, we can't recommend it to users for this reason. If not for the Quick Introduction Guide, *LiVES* would suffer a similar fate.

Openshot and *Shotcut* faired nearly identical in all the tests. The use of images to explain the different effects by the latter is a novel feature, which all the other tools should consider adopting. We've very reluctantly pushed *LiVES* off the podium on account of its lack of official binaries for different distributions. LXF



1st Kdenlive 9/10

Web: https://kdenlive.org Licence: GPL
Version: 18.04.1
Its interface is cluttered, but the rich documentation makes it easy to use.

2nd Openshot 8/10

Web: www.openshot.org Licence: GPLv3+
Version: 2.4.2
Designed especially for animations, but works just as well for regular videos.

3rd Shotcut 7/10

Web: https://shotcut.org Licence: GPLv3
Version: 18.09
Doesn't take long to start producing sharp content.

4th LiVES 6/10

Web: http://lives-video.com Licence: GPLv3+
Version: 2.10
Feature-rich but far slower than the others.

5th Flowblade 5/10

Web: https://jliljebl.github.io/flowblade Licence: GPLv3+
Version: 1.16
Only for users who are already familiar with video editing.

» ALSO_CONSIDER

A quick look at the Wikipedia page for video editors will reveal close to a dozen active open source applications. Couple this with all the freeware offerings and you get a sizeable number of alternatives to choose from.

Cinelerra is a professional-grade editor that's complex for absolute beginners. *Avidemux* is fairly straightforward to use, but only provides basic functionality. It's also not as actively developed as the tools featured in this *Roundup*.

While we've focused on the more established applications in this *Roundup*, *Natron* is a viable alternative. First released in 2014, its interface is influenced by commercial professional-grade software such as *BlackMagic Fusion* and *Nuke*.

Although it hasn't seen a new release since 2016, *Pitivi* is another alternative. Available as a Flatpak package, the application was once tightly integrated into Gnome desktop, but now works flawlessly with all desktops.

Roundup

Google Drawings » Gravit Designer » Inkscape » LibreOffice Draw » Vectr

Mike Bedford is always on the lookout for ways to do things differently.

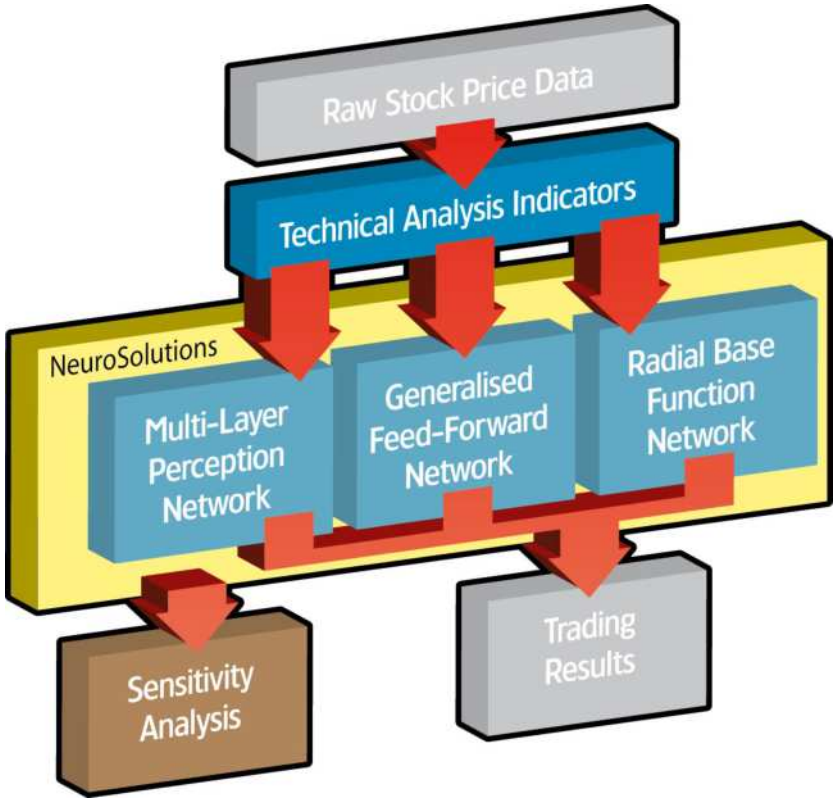
Vector graphics

Vector-based drawing software is ideal for creating technical and business diagrams and illustrations. **Mike Bedford** put five packages to the test.

HOW WE TESTED...

The most obvious differentiator between vector drawing packages is their more advanced graphics features, so it could be argued that any testing strategy should concentrate of these. However, a significant majority of users have little need for many of the more esoteric features; what they really need are just the basic tools for creating illustrations and diagrams for technical, scientific or business applications. This, therefore, has been our emphasis.

That doesn't mean, however, that all software is equal. While the basic drawing tools might be largely similar, products differ significantly in areas related to productivity, so our testing has also considered this aspect. This goes far beyond the design of the user interface. Included under the heading of ease-of-use, for example, is the availability of tutorials and good user documentation, while in a team scenario, features that simplify collaboration and sharing of illustrations are vital. We've also looked at licensing.



So-called paint packages are the most commonly used type of graphic editing software. But while they're perfect for touching up photos, they're far from ideal for creating diagrams and illustrations. The snag with using paint software for this purpose is that they work at the level of pixels. A much better solution is to use a drawing package, otherwise known as vector-based graphics software, which works at the level of picture elements such as lines, circles, boxes and text. The waters have been muddied by some paint software appearing to offer similar functionality, but if you want full control over the drawing process, there's no better option than vector-based software.

This type of software can used by graphics artists, but our assumption here is that most **LXF** readers will have more down-to-earth applications, for which vector-based software is ideal. Perhaps you're a scientist or engineer with a need to create 2D technical drawings or diagrams. Maybe you need to produce flow diagrams. Or conceivably you have a business application such as creating a company logo or drafting an organisational diagram. For all this and more, there's something to meet your needs.

Ease of use

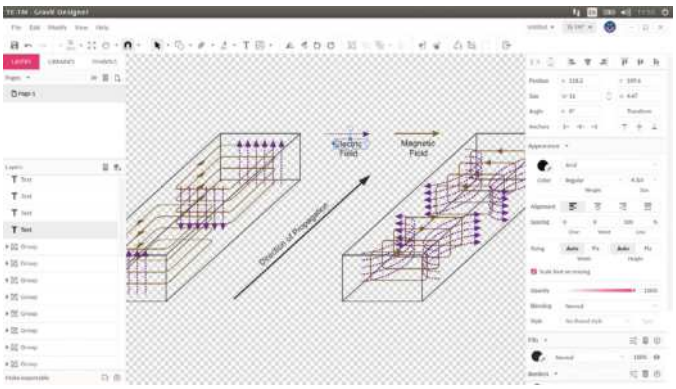
Spotlight on productivity.

Except in the case of a very poorly designed interface, where commonly used functions might be several levels down in the directory structure and there are no icons to act as shortcuts, regular users will eventually get up to speed with any software. Admittedly, you'll learn some packages more quickly than others, but there's more than an element of truth to the suggestion that it's the software with fewer features that will fare better in this respect.

For these reasons, our appraisal of ease-of-use has considered a lot more than a potentially subjective assessment of the user interface. In particular, and especially for the more fully-featured packages, we have given consideration to resources such as the user documentation and tutorials.

We found that we could use *Google Drawings* almost immediately without referring to Help, but primarily that was because it's by far the simplest in terms of its functionality. All the others, by necessity, have a more complicated user interface because of their richer set of features. Having said that, despite it probably being the most fully featured, we didn't find *Inkscape* the hardest to use, but we recognise that this view might be somewhat subjective.

Turning to educational and reference material, all the software has official documentation in the form of a user guide, while *LibreOffice Draw* also offers a Getting Started guide. Documentation differs significantly in how extensive it is, with the *Google Drawings* guide being much shorter than the others, although this is hardly surprising, given its fewer features. By way of contrast, the *Inkscape* website refers to several manuals and



Gravit's modern user interface has drawn much praise.

books, albeit not all free. All the packages except for *Google Drawings* also provide online tutorials, mostly textual, although *Gravit* has no fewer than 13 official YouTube video tutorials. *Vectr* has a particularly impressive set of 29 text tutorials, some concentrating on features, others on common applications.

So far, we've referred only to official user documentation or, in some cases, third-party documentation that is officially sanctioned by being linked from the publisher's website. You'll find a wealth of other third-party video guides and tutorials for all the software, even *Google Drawing*, but we haven't reviewed these because, by their nature, they're likely to come and go.

VERDICT

GOOGLE DRAWINGS	7/10	LIBREOFFICE DRAW	6/10
GRAVIT DESIGNER	8/10	VECTR	8/10
INKSCAPE	8/10		

On balance, ease-of-use is good across the board.

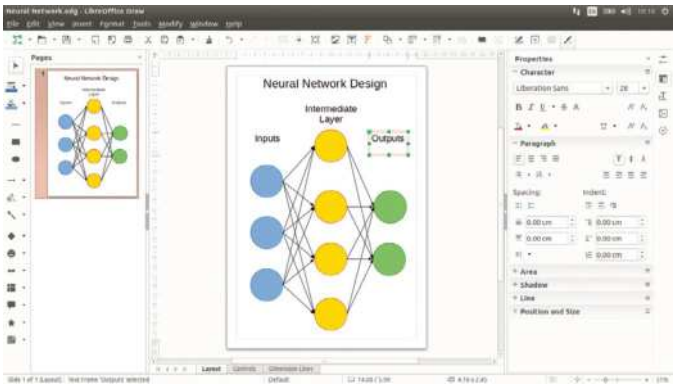
Licensing

Examining the small print.

All our chosen software is free. This choice was made partially because the highly respected *Adobe Illustrator* and *CorelDRAW* aren't available under Linux, but also to illustrate the fact that you can get performance comparable to these leading commercial packages without spending several hundred pounds. In other respects, though, there are differences between our five packages.

First of all, they're not all open source which, while probably not deterring users who just need to create good quality graphics, will be an issue for more technically minded users with more ambitious aims. *Inkscape* and *LibreOffice Draw* are open source, the others are not. In addition, in order to use any of the three which are not open source you'll need to create an account with the publisher. In the case of *Google Drawings*, that's just a general Google account which you might already have.

The situation with *Gravit* might be a concern to some potential users. The product has now been sold to Corel and, while the basic version of *Gravit Designer* remains free to use, it has now been joined by *Gravit Designer Pro* (for which you get a 14-day free trial when you sign up) that costs £75/$99 per year. Gravit say it has no plans to discontinue the free version, but questions remain. For example, might some of the features of the free



By no means universal among drawing software, LibreOffice Draw has an open source license.

version become available only in the *Pro* version – or will new features only be offered if you're prepared to pay for them? It undermines confidence in the long-term functionality.

VERDICT

GOOGLE DRAWINGS	7/10	LIBREOFFICE DRAW	10/10
GRAVIT DESIGNER	5/10	VECTR	7/10
INKSCAPE	10/10		

Only two packages are open source and, while all are free, Gravit has commercial aspirations.

Drawing tools

Investigating the core functionality.

There can be no more important features of a vector-based drawing package than the drawing tools it provides. In this paragraph we define the basic minimum, as provided by all our software. Tools are provided for drawing lines, lines with arrow heads, text, and shapes such as squares and circles, and to define the thickness and colour of the lines or outlines, and the colour of any fill. In addition to straight lines and polylines – that's a line comprising several straight segments – freehand and curved lines are available.

Editing enables objects to be moved, stretched or shrunk, and rotated. A snap feature assists in aligning objects, and the order of objects, front to back, can be defined. Several objects can be grouped so they can more easily be manipulated. Groups can subsequently be ungrouped, if necessary, for manipulating their individual constituent objects.

Google Drawings 5/10

Google Drawings offers the basic drawing tools and little more, but just saying that doesn't do it justice. For example, while some drawing packages offer just a handful of shapes – commonly rectangles, rectangles with rounded corners and circles – Google Drawings offers a vast number, under the headings of shapes, arrows, callouts and equation. This emphasises the most commonly suggested application for the software, namely the creation of flowcharts, block diagrams, organisational charts and so on.

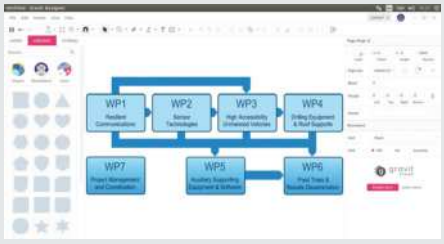
Text support is also good which makes this solution suitable for simple design applications such as the creation of flyers containing a mixture of text and graphics. Basic word art is offered in addition to text boxes, the latter offering the sort of formatting options found in word processors. More usefully as number of template diagrams are offered.

Gravit Designer 8/10

Promotional literature for *Gravit* commonly shows artistic design applications, which hints at the fact that it has more advanced drawing facilities than most vector editing packages. However, we're of the view that the more sophisticated features won't normally be needed for technical applications, so they'll generally be of interest only to more imaginative illustrators.

Gravit offers nearly everything provided by the other packages reviewed here, with the exception of connectors and dimensioning, but adds a range of extra facilities. Of particular note are master pages and styles, which can offer major benefits in improving your productivity.

Most of the other advanced features will be of relevance to a minority of readers, but the key add-ons include a knife tool, non-destructive Booleans, path graphs and advanced colour support.



Import and export

Which file formats are supported?

Realistically, the only reason to create a diagram or illustration is so that you can display it online or insert it into a document, so it's essential that you can save or export it in a suitable format. Similarly, if you want to work on a drawing someone has created or, perhaps, include a third-party image in your drawing, the range of supported import formats becomes important.

Nearly all the reviewed software uses a proprietary format for saving drawings, which isn't too helpful – except in the case of *LibreOffice Draw*, because its native format is recognised by other *LibreOffice* applications. The one package that can save in a variety of formats is *Inkscape*, which supports SVG, compressed SVG, PostScript/EPS, *Adobe Illustrator* and several other vector formats. Separate from the save formats, most of the software can export in several file types, including bitmapped as well as vector formats. Bear in mind, though, that if you export in a bitmapped format without also saving it in a vector format separately, you won't be able to edit it properly afterwards.

Inkscape supports the widest range of import formats, including just about anything you'd ever want. However, all offer import of at least a few bitmapped formats, and most also allow you to import the vector SVG format. *Google Drawings* is particular poor in this respect, though; this may be due to the fact that it's an online-only tool, working through Google Docs. We didn't find a list of supported import formats, and our tests didn't reveal any vector formats that worked. We did find reference to a few workarounds suggested online, but didn't manage to get any to work either.

VERDICT

GOOGLE DRAWINGS	2/10	LIBREOFFICE DRAW	5/10
GRAVIT DESIGNER	7/10	VECTR	5/10
INKSCAPE	10/10		

Somewhat surprisingly, there is a vast difference in the number of file formats supported for import and export.

Inkscape 10/10

Inkscape is commonly compared favourably to *Adobe Illustrator*, a fully featured commercial offering with a hefty annual licence fee. It has an impressive range of features which means that, like *Gravit Designer*, it will appeal to graphic designers as well as those with more down-to-earth applications.

We're not aware of any major drawing tools provided by any of the other software here that aren't also in *Inkscape*. The dimensioning feature for engineering drawings is provided by an extension, but this isn't a major drawback; indeed the support for extensions is one of *Inkscape*'s significant features.

The list of advanced features is long; some of the most significant include support for generalised polygons and stars, spirals, Boolean operations, and calligraphy tools for freehand drawing using filled paths.

LibreOffice Draw 7/10

Compared to our basic set of drawing tools, *LibreOffice Draw*'s main extensions are in the areas of layers and pages, 3D objects, connectors, and dimensioning. Its support for 3D objects is provided primarily by allowing 2D objects to be extruded, although a library of ready-made 3D objects is also available. Once you have a 3D object on-screen, you can apply various effects like lighting and shading.

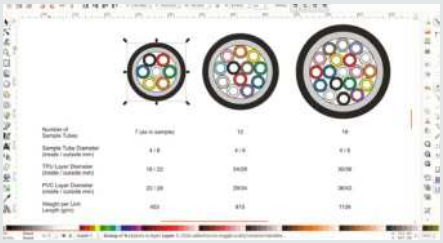
The concept of connectors very much simplifies the editing of diagrams such as flow charts or organisational charts, where boxes are connected by lines. When you connect objects using a special type of line called a connector, those connecting lines stay connected to their associated objects. If you move an object, its connectors remain connected and re-route accordingly. Finally, automatic dimensioning labels are valuable for engineering applications.

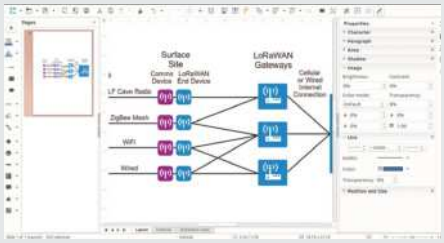
Vectr 6/10

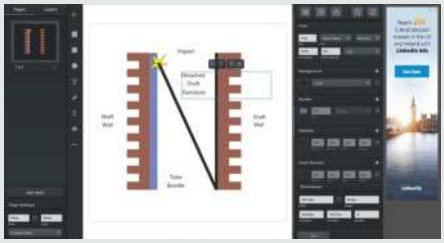
Vectr offers more than basic drawing tools, but not by a huge margin. First, there's support for layers and pages. Layers enable you to assign objects to different layers, which can then be dragged above or below other layers, locked or hidden.

The concept of pages is fundamental to word processors, but it's not universal in the realm of graphics editors. However, while it means that you can keep all the elements in a multi-page element together – unlike some of the other packages reviewed here – it doesn't provide the full benefit you might expect, because you can only export one page at a time.

Another interesting feature is filters. For example, you could define both a solid colour and a graduated fill as an object's background with different opacities, although this won't be on most people's must-have list.







Bitmap support

Going beyond just vector graphics capabilities.

Although we're looking at vector-based graphics packages instead of the bitmapped software used for editing photos, there are times when some limited support of bitmaps is useful, say for including photos of staff in an organisational chart. All the software reviewed here enables you to import bitmapped images for this purpose, and the basic editing functions available enable you to move, resize and rotate them. Some software also enables you to crop images but, even if there's no specific crop bitmap function, with most reasonably featured vector graphics software you'll be able to find some way of achieving this.

While we wouldn't recommend using any vector-based software for dedicated image editing use – something like *GIMP* is far more appropriate – some of the packages included here do offer a bit more than the basics in terms of support for bitmaps and sometimes a lot more. This might streamline your workflow by not requiring you to edit images in other software before importing them to your vector-based editor. *Gravit* is particularly strong here, enabling you to adjust brightness and contrast, and even fine-tune colours and remove noise. While *Inkscape* doesn't offer all this natively, the raster extension adds all *Gravit*'s bitmap editing capabilities and a lot more. *LibreOffice Draw* also has advanced features in this area – mostly artistic effects.

Another useful feature associated with bitmaps is to convert them to vectors. This is variously called tracing, vectorising, or converting to polygons. Once converted to vectors or polygons, all the usual vector editing facilities become available. This functionality is available in *Inkscape*, *LibreOffice Draw* and *Gravit*.

VERDICT

GOOGLE DRAWINGS	4/10	LIBREOFFICE DRAW	6/10
GRAVIT DESIGNER	8/10	VECTR	4/10
INKSCAPE	10/10		

All packages offer some sort of support for bitmaps, but some go well beyond the basics, with more advanced features.

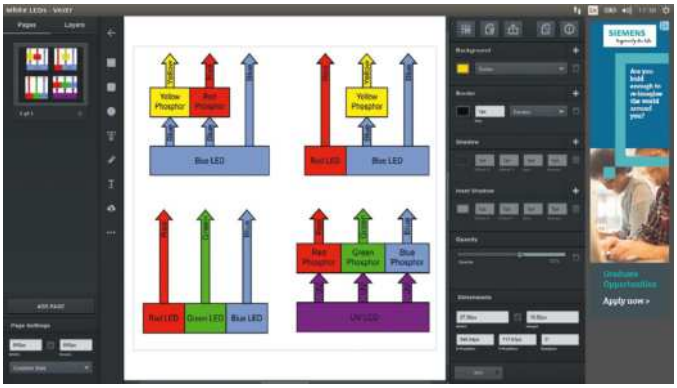
Online/offline

Do you need an internet connection to run it?

Google Drawings is available only as an online application, in the sense that it runs in your browser. Conversely, *Inkscape* and *LibreOffice Draw* are only available to run locally on your PC. There is an application called *LibreOffice Online*, but this is provided only as software and not as an online hosted service. *Gravit* and *Vectr*, on the other hand, can be run either online or offline. Strictly speaking *Gravit Designer*, as reviewed here, is for offline use only, but the same publishers also provide *Gravit Cloud*, which provides the same features, but which runs in the cloud.

Both approaches have their supporters, but rather than automatically being drawn to the one solution, it would be appropriate to consider the pros and cons of having just an online or just an offline solution. Clearly if you only have online drawing software, you can't use it if you're without an internet connection – perhaps when you're on a plane, or don't want to have to pay for Wi-Fi.

You might think that the offline option is therefore preferable, but this is only true if you have a PC on which it's installed to hand. With online software, so long as there's an internet connection you could work from any PC, running under any operating system for that matter. The dual approach, therefore,



Vectr might offer online and offline version but you need to be online to use the offline version.

seems to offer a win-win solution. This might seem to favour *Gravit* and *Vectr*, but things aren't quite that simple with the latter because, even if you have it installed locally on your PC, you will only be able to use it if you log in via an online connection – one of those 'features' than benefit the publisher rather than the user.

VERDICT

GOOGLE DRAWINGS	6/10	LIBREOFFICE DRAW	6/10
GRAVIT DESIGNER	10/10	VECTR	6/10
INKSCAPE	6/10		

Gravit is unique in offering both online editing and a complete offline offering, while the others vary in capabilities.

Collaboration

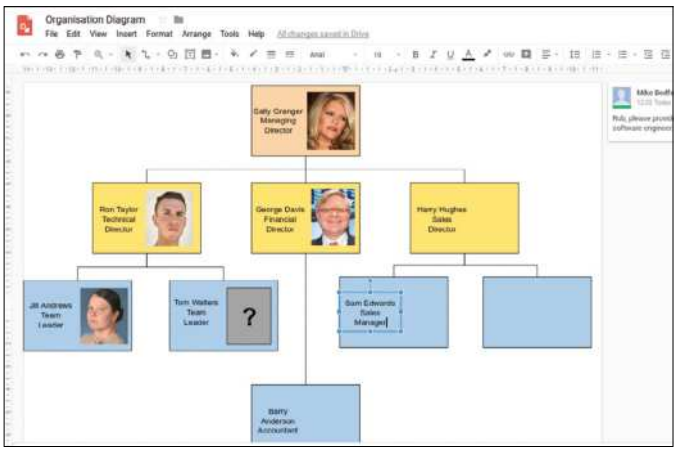
Helping you work as a team.

When you're working as part of a team, perhaps in business or as part of a club, being able to collaborate with others will sometimes be necessary. There are several aspects to this, some of which we've given consideration to already. Even if your collaboration involves nothing more sophisticated than emailing files back and forth, or sharing them via an online file sharing service, an extensive set of supported vector-based file formats for import and export is useful. This might permit you to collaborate with people who are using different software.

All the packages have at least one vector-based format for import and export, with the exception of *Google Drawings*. However, this doesn't mean that Google is inappropriate for collaboration, as we're about to see.

Another way of ensuring collaboration is to insist that all members of the team use the same software. If your co-workers use a variety of operating systems, however, this could be difficult. *Inkscape* and *LibreOffice Draw* run under Windows and macOS as well as Linux, *Vectr* runs under Windows and ChromeOS, and Gravit runs under all three. This question doesn't apply to the Google offering since it's accessed online.

Going beyond this, some software has specific facilities to enable collaboration. As an online utility, *Google Drawings*, for example, permits several users to work simultaneously on the same drawing. *Vectr* says it is working on a similar facility but, in the meantime, and bearing in mind that *Vectr* always saves your work online even if you're using the offline version, you can share



If you want to work as part of a team, Google Drawings is ideal.

your work with other users. Despite the existence of *Gravit Designer*'s online stablemate *Gravit Cloud*, *Gravit* is one step further behind in offering collaboration, saying only that it's under consideration. The online *Gravit Cloud* does streamline the process slightly since a team could share the same account. Needless to say, in such a scenario, you'd have to ensure that multiple people don't try to edit a project at the same time.

VERDICT

GOOGLE DRAWINGS	10/10	LIBREOFFICE DRAW	5/10
GRAVIT DESIGNER	7/10	VECTR	7/10
INKSCAPE	6/10		

As you'd expect from an online-only app, Google Drawings is unparalleled in its support for collaborative simultaneous editing.

Vector graphics
The Verdict

Initially it wasn't immediately obvious to us which of our five products would take the top spot – after all, they each have their pros and cons – but averaging the scores for each of our tests gives an unequivocal answer. Note that, while three products are shown with equal scores of 6/10, their order is definitely as shown, because the scores all differ when worked out to reveal fractional values.

While the combined scores favour *Inkscape*, which is slightly ahead of *Gravit Designer*, it's not necessarily the best product for everyone because different users have different needs. If we look exclusively at the drawing functions provided, the top two spots are the same as the averaged rankings, with *Inkscape* the clear leader and *Gravit Designer* second.

However, this alone shouldn't be your primary consideration unless you want to go beyond technical diagrams or business charts. If you are interested in more artistic use, advanced features will be important, otherwise different aspects will come to the fore.

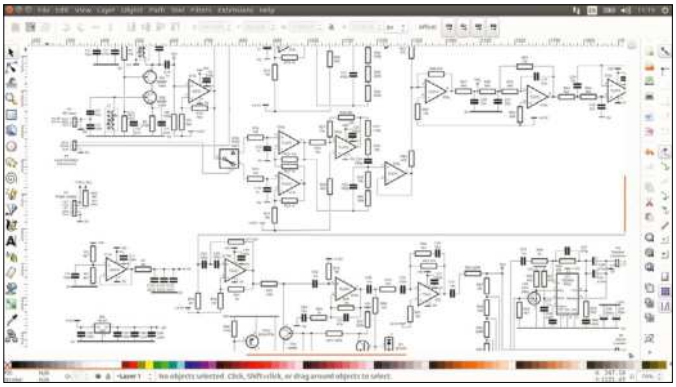
Productivity is, of course, particularly important, especially for business users. However, aspects such as ease-of-use and the provision of user documentation and tutorials, when taken together, don't suggest a clear leader in this area.

Licensing will concern some potential users and, while all are free, only *Inkscape* and *LibreOffice Draw* are open source. There is also the question over the future of the free version of *Gravit Designer*, given that it has now been joined by the commercial *Pro* version.

On the subject of productivity, the availability of both an online and offline version can help but, while *Gravit Designer* and *Vectr* offer both, a locally installed copy of *Vectr* can only be used with a connection to the internet.

Collaboration will be relevant to some users, and packages differ in how well they support this by supporting a wide range of import and export file formats, and being available on several operating systems. Here *Inkscape* is head and shoulders above of the others.

In terms of specific support for collaborative editing, though, *Google Drawings* is the clear winner, despite its very basic drawing facilities falling somewhat behind the competition.



1st Inkscape 9/10

Web: inkscape.org Licence: Open source
Version: 0.48
Unparalleled for its advanced features, albeit more than many users will need.

2nd Gravit Designer 8/10

Web: gravit.io Licence: Closed source
Version: 3.5.6
Excellent all-rounder but there are concerns about the free version's lifespan.

3rd LibreOffice Draw 6/10

Web: www.libreoffice.org/discover/draw Licence: Open source
Version: 6.0.7
Ideal for most technical applications and pre-installed with many distros.

4th Vectr 6/10

Web: vectr.com Licence: Closed source
Version: 0.1.16
Reasonable overall, with just the odd drop-off.

5th Google Drawings 6/10

Web: docs.google.com/drawings Licence: Closed source
Version: Not specified
Ideal for teams due to its support for real-time collaboration.

» ALSO_CONSIDER

For general purpose use, we find it hard to believe that one of the packages reviewed here won't fit the bill, but if you do want to delve a bit further, other free options – which either run under Linux or are accessed via a browser – include *Dia*, *Method Draw*, *sK1* and *SVG-edit*.

If you have a very specific application, such as creating flow diagrams, organisational diagrams and the like, there are some other programs you might like to consider. While these won't have nearly as many features as most of the software we've looked at here, the features they do have are tailored to their niche applications, which makes them much easier to use. *Diagramo*, for example, calls itself a flowcharting program, although the samples show other types of block diagrams. Other packages intended for various types of charting applications, often described as being like *Microsoft Visio*, include *Calligra Flow*, *Dynamic* and *yEd Graph Editor*.

Roundup

Gimp » Rawtherapee » Darktable » Lightzone » Pixeluvo

Alexander Tolstoy likes nothing better than rolling up his metaphorical sleeves and seeing what new gems the open source community has to offer.

Photo editors

Keen photographer **Alexander Tolstoy** keeps his imperfect snaps, for fixing later, using some of the best photo editors available for Linux.

HOW WE TESTED...

Editing photos isn't the same as editing general images. Therefore we decided to compare some of the best photo editors for Linux with a focus on fixing common faults, enhancing colours and lost details that might be considered gone for good.

The aim here is to help you select the most feature-packed and practical software that can bring dull images back to life. We'll take a closer look at their advanced features, such as automatic enhancements, the ability to fix dull, blurry, under- or over-exposed images and, of course, whether or not they handle RAW files.

It's not the first time we've run a *Roundup* on image or photo editors, but this current group includes a different selection of software. Four out of the five are open source tools (although *Lightzone* had been a proprietary application in the past). The fifth is a commercial option, which is perhaps less known to the reader, but should be of great interest to photo-editing enthusiasts. Read on to find out more!

When it comes to selecting a photo editor, there's plenty of choice. There are general-purpose editors like *Gimp* or *Pinta*, image library managers with editing features, such as *Digikam* or *Shotwell*, and finally there's software for digital artists (think *Krita*). Our interest lies in fixing photos and processing RAW files, so we grabbed *Gimp* with its set of built-in effects, UFraw plugin, and lots of third-party plugins from FX Foundry and G'MIC. Then we included *Darktable* and *Rawtherapee* – two classic darkroom editors with tools for dragging out barely visible details. *Lightzone* is similar to these two programs, but has a more consumer-friendly interface. Last but not least is *Pixeluvo*, a paid-for program with a limited set of editing features.

Gimp is the only contender that doesn't use a dark UI theme by default. Although a dark theme adds a 'pro' feel to the program, it won't hide the truth in our *Roundup*!

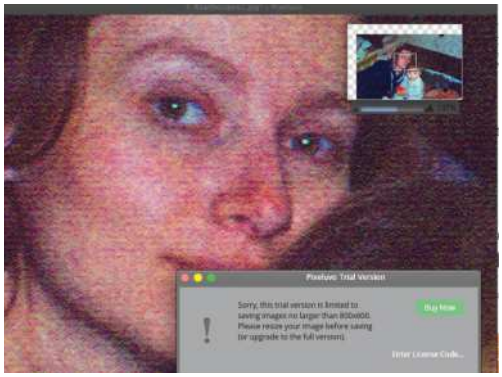
Availability

How to get a program working...

Sometimes it requires extra efforts to obtain and install a program. You might have to visit its web site and find the correct download option. There are no such issues with *Gimp* because it's included with every Linux distribution, but we're working with a supercharged *Gimp* installation, packed with extra plugins. GREYC's Magic for Image Computing might be available in your system's package repositories, but if not you can grab it from **https://gmic.eu**, unpack the archive and put the executable file in **~/.config/GIMP/2.10/plug-ins**. The FX Foundry pack has its own page at **www.gimpfx-foundry.sourceforge.net** and similarly it should be unpacked into a neighbouring directory in **~/.config/GIMP/2.10/scripts**. Although there's nothing difficult about this requirement, novice users may consider such extra actions to be an advanced task.

Rawtherapee and *Darktable* already come with everything bundled inside the software packages. Those two programs are rarely included in distributions' default offerings out of the box, but they're always available in online repositories – just check it with your software store or package manager. However, *Darktable* scores more in this test because it's also available in Snapstore and Flathub, while *Rawtherapee* has only an AppImage.

The way *Lightzone* is distributed is somewhat cumbersome. Due to the proprietary legacy, the official project web site still lists outdated information, which might confuse some users who want to get the application. However, *Lightzone* is based on Java, which means it should be a platform-agnostic application, and so it's often packaged for many Linux distros. Ready-to-use packages exist for Ubuntu, Fedora and openSUSE and many others. For Ubuntu-compatible distros you can use the PPA at **ppa:lightzone-team/lightzone**.



Pixeluvo includes a functional limitation, which is supposed to motivate users to purchase the full version.

Pixeluvo costs money, $34 to be precise. The official web page offers 30-day trial downloads, with DEB and RPM packages for Ubuntu and Fedora, respectively. Those packages can be used for other RPM- and DEB-based distributions as well, or repacked for other flavours of Linux (treat *Pixeluvo* downloads like archives). *Pixeluvo* includes a static set of the old Qt4 libraries at **/opt/pixeluvo/libs64**, which you may want to delete in order to force the program to use your system's libraries. We'll admit that this is a non-intuitive task for many users, Additionally, the trial version doesn't enable saving modified images with sizes larger than 800x600. We recommend making those tweaks!

VERDICT

GIMP	9/10	LIGHTZONE	8/10
DARKTABLE	10/10	PIXELUVO	7/10
RAWTHERAPEE	9/10		

There are many easy methods of getting hold of and installing Darktable.

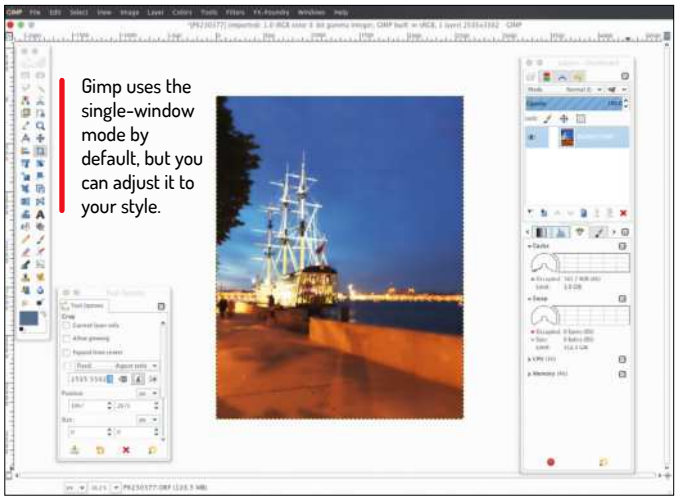
Ease of use

How user-friendly is each of our five contenders?

Previously, *Gimp* was a mess of floating windows that required a lot of manual arrangement. Nowadays the program uses the single-window mode by default. Its photo-editing features are located in different places under Filters>Enhancement, Filters>G'MIC-Qt and a separate top-level menu for FX Foundry. Unless you know where your tools are beforehand, it isn't easy to find them.

Darktable has no such fragmentation. However, you first need to point the application to the directory with your images and then select one for editing. By default, the editing tools are hidden under the categories on the right side of the screen. *Rawtherapee* has a more thoughtful UI design. It's similar to the one in *Darktable*, but you don't lose the file manager tree on the left when editing an image. Furthermore, all the useful tools are already expanded in the right panel. Similar to *Darktable*, *Lightzone* suggests switching between browsing and editing modes. When editing, each filter or tool is presented like a separate floating panel, which you can turn off or on.

Our remaining program, *Pixeluvo*, also shone in terms of usability. Photo-editing capabilities could be found under the



Gimp uses the single-window mode by default, but you can adjust it to your style.

Colour and Effect sections of the top menu bar. *Pixeluvo* was designed with simplicity in mind, and consequently it was nearly impossible to get lost in its clear and simple lists of available tools.

VERDICT

GIMP	7/10	LIGHTZONE	10/10
DARKTABLE	8/10	PIXELUVO	10/10
RAWTHERAPEE	10/10		

Having frequently used tools at hand is the key to winning our usability test.

Fixing colours, shadows and more

Want to give your photos a natural look?

We may prefer one tool over another purely for subjective reasons, but there are some practical tests that expose the real value of each photo-editing program.

For this *Roundup* we used a couple of images as test subjects when exploring the five programs. One was a scanned historic colour film frame with inadequate lighting, film grain and insufficient exposure, while another was an unprocessed RAW shot of a night urban scenery with illumination, which was also underexposed.

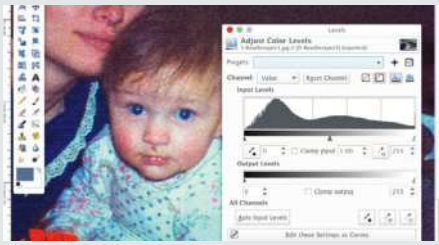
The aim was to try to achieve the best possible results from both shots in each of our five programs. We paid special attention to the white balance fixer and any available colour adjustment sliders, as well as the shadows, highlights and level correction tools.

In the end, a good program is one that can either fix everything using automatic algorithms, or by offering adequate tools for manual work. It goes without saying that the only acceptable editing mode is a non-destructive one, where the user can always roll back to the original image without losing anything irreversibly.

Gimp 10/10

When it comes to correcting colours, *Gimp* shines thanks to its fast GEGL engine. We didn't need any third-party plug-ins for fixing the colours of our test photo. The Colour>Levels menu enabled us to fix the white balance (look for three pickers for choosing black, grey and white spots), but we also managed to remove that distasteful green hue on certain parts of the image by adjusting levels manually for green channel only. Using *Gimp's* standard features for tuning up levels and curves turned out to be easier than in specialised software such as *Rawtherapee* and *Digikam*. Not that one application is better than another, but *Gimp* is just easier to use for an average user.

When processing RAW files, *Gimp* can integrate with *UFRaw*, *Rawtherapee* and possibly other similar tools. The result depends on how well a user can handle the many tools of a RAW processor.



Darktable 6/10

Darktable has some advanced tools for working with colours, including a dedicated White Balance section. The neutral Color Picker tool sits under the Levels section, but it's not intuitive and takes quite a lot of time to fix white balance. Getting this tool to work correctly in *Darktable* is crucial. We achieved acceptable results by adjusting the colour temperature and each of RGB channels separately. We believe that *Darktable* has too many tools, many of which overlap, so the program is more likely to help you achieve artistic effects than bring back natural colours.

The positive side of *Darktable* was working with RAW files. For example, the application offers the original algorithm for exposure compensation, which resulted in a pleasing image with less noise and better local contrast. Another *Darktable*-only tool is Velvia, which boosts image saturation for brighter parts of the image only.

Removing image noise

How good are these programs for eliminating noise and restoring detail?

The reason why we equipped *Gimp* with third-party plug-ins is because they grant the program advanced tools for combating colour noise and removing grain. The Repair category inside the G'MIC interface contains some advanced denoising filters, of which the most powerful is perhaps Iain's Noise Reduction. This filter contains a range of configurable options that effectively remove film grain or colour noise that has been created by high ISO values.

We managed to achieve reasonable results simply by combining several filters in succession. However, the built-in options for reducing noise and grain in *Darktable* proved to be even more efficient because they managed to preserve details more effectively. *Darktable* has a bilateral denoise feature for high ISO; the non-local tool can be used when denoising chroma and luma; there's a dedicated chromatic denoise tool for RAW images; and the universal Equalizer tool can enhance many things, including clarity as well and reduce colour noise. *Rawtherapee's* denoising tools are slightly less effective, but when it comes to processing RAWs with high ISO, we think *Rawtherapee* has the edge because it produces much smoother JPEGs compared to those processed by *Darktable*.

In contrast to the other programs, *Lightzone* just has a simple denoise tool with only two sliders for colour and grain. Sadly, for some of our test images, that denoise filter had little or no effect.

Finally, the Reduce Noise feature in *Pixeluvo* is only there for the sake of box-checking. It works as a primitive median filter and removes noise together with precious details. It's best avoided unless you want to change the nature of your original photo.

VERDICT

GIMP	8/10	LIGHTZONE	5/10
DARKTABLE	8/10	PIXELUVO	5/10
RAWTHERAPEE	10/10		

Choose Gimp for retouching old photos and go with Rawtherapee for creating perfect JPEGs out of your RAW shots.

Rawtherapee 10/10

We started with our greenish old film and tried to fix its colours in *Rawtherapee*. First we rolled out tools under the Colour section, chose the white balance picker and quickly made the colours look much more natural. However, we also had to manually remove the erroneous green hue of the wooden bar while keeping the rest of the image natural. *Rawtherapee* has a rich set of colour correction tools, including colour channels and RGB curves for selective colour correction.

For the second image we switched to the Exposure section, hit the Auto levels button and then adjusted colour compression, brightness, contrast and exposure sliders to make the picture brighter and more vivid in general without turning illuminated parts into glowing spots. It wasn't too difficult thanks to *Rawtherapee* providing all the necessary tools for achieving this.

Lightzone 10/10

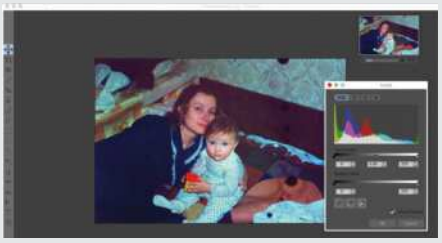
Playing with colour settings in *Lightzone* proved to be relatively easy. The application offers its own original way of adjusting colour balance, which was a bit unusual but no less efficient in the end. Expanding the White Balance section in the Edit mode enabled us to select a neutral spot in the image, making colours look more natural. In addition, the Color Balance feature provided us with a tool to select a colour range and change its temperature. This approach was perhaps the best and the most intuitive for our purposes, when we needed to selectively remove certain tints from the image.

Working with dark night shots led us to the Relight section, a dedicated tool set for repairing underexposed parts of the image. Another useful thing was Zonemapper with its luminosity gradient. This tool changes the exposure value, whereas Relight sliders are ideal for shadows and highlights.

Pixeluvo 10/10

There are plenty of colour-related tools in *Pixeluvo*, including Color Balance, Replace Color and Levels sections for working with colour temperature and white balance. The neutral colour picker is also there, under the Levels settings. *Pixeluvo's* available tools show a noticeable shift towards creative image processing, such as toning, colourising, boosting vibrance and improving local contrast. The number of tools is limited, but everything you need is here. It'll take you a little time to learn how to fix images with incorrect colours in *Pixeluvo*, but if you decide to buy the software your money will be well spent.

Pixeluvo includes a decent RAW processor with some useful features. There's the Auto-exposure button (worked like a charm for our test image) and a bunch of sliders including the useful Recovery option. *Pixeluvo* worked perfectly when enhancing RAW image files, too.



Editing tools

How about brushes, layers or selection tools?

Gimp shines in this test, because the program was created exactly for image manipulation of this kind. *Gimp* has the gorgeous Healing Brush tool for super-smart removal of any spot defects present within a photograph. Needless to say, *Gimp's* selections, masks and layers that you can group or rearrange is something that any photo-retouching specialist frequently uses.

Darktable has a useful spot removal tool, which also enables you to add circles, ellipses and custom outlines to your photographs. When used in conjunction with Masks, this makes it possible to repaint custom areas of the image, but it's a far less intuitive process compared with using similar image-editing tools that live within *Gimp*.

Speaking about *Lightzone*, there's a roundabout way of working with layers within the program. Most filters in *Lightzone* have the Colour Selection tab that enables you to select a range of colours, for which the filter will be applied. Together with the Colour Mask view mode this works like the layer approach. Each filter is therefore a separate layer of the source image. There's also a spot removal tool.

The *Pixeluvo* editor has a far better set of editing tools and includes such cool features as Warp, FX brush, Clone brush, Straighten tool and many more. The lower right part of the image has a retractable layer panel with all the basic features that a good image editor should have. A very pleasant experience!

Unfortunately, *Rawtherapee* doesn't offer any means for working with parts of an image. The best way to solve the problem would be to use an external editor (any other in our *Roundup*).

VERDICT

GIMP	10/10	LIGHTZONE	8/10
DARKTABLE	6/10	PIXELUVO	9/10
RAWTHERAPEE	1/10		

Rawtherapee only works with entire images. For editing tools we'd recommend choosing either Gimp or Pixeluvo.

Special features

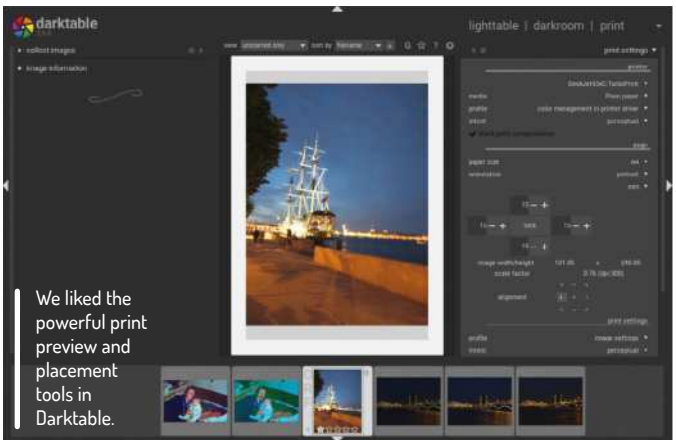
How much extra value can you get with one program?

We also tested features that didn't fall into any of the previous sections. One of *Gimp's* most valuable feature is its extensibility. There's a range of useful extra editing scenarios and plug-ins for *Gimp*, from motion blur remover (Refocus) to some great photo filters from FX Foundry (see Photo>Enhancement). It turns out that *Gimp* is the most versatile and at the same time feature-rich image editor for Linux.

Darktable has enough benefits too, from OpenCL support (*Gimp* has this, too) to easy lens profiling, keywords editor and print preview feature. *Rawtherapee* has few extra options, but we really liked its powerful file browser that can carry out batch processing and apply edit profiles to several images at once; this feature in *Rawtherapee* is much easier to use than copying and pasting history stacks in *Darktable*!

Lightzone contains a beautiful set of ready-to-use presets called Styles. Each style is a preconfigured existing filter or effect, which you can always fine-tune to your liking. The way *Lightzone* effects are stacked is perhaps the best one in terms of usability. You can have as many effects applied to the image at once as you like, with one of the best effects being Relight.

Moving on to *Pixeluvo*, we enjoyed the Quick Color feature, which is basically a set of one-click Instagram-like effects. The



We liked the powerful print preview and placement tools in Darktable.

Warp tool was also a unique tool for creating instant and fun distortion effects. While definitely not a tool for professionals, *Pixeluvo* contains all the necessary functions for an average photo processing and retouching pipeline, and for many people this would be sufficient.

VERDICT

GIMP	10/10	LIGHTZONE	7/10
DARKTABLE	8/10	PIXELUVO	9/10
RAWTHERAPEE	7/10		

Despite the handy Spot tool and layers, Lightzone doesn't have many extra features compared with the other programs on test.

Help and support

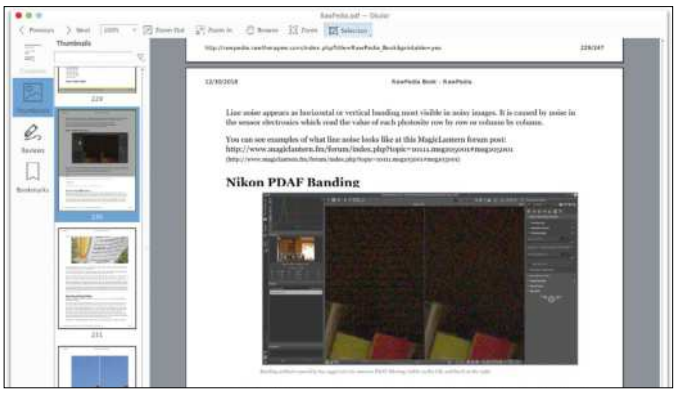
Can you become a pro user easily?

Some time ago *Lightzone* insisted on users registering their data on the website before obtaining permission to download the application. This is no longer an obligatory step, but we understand that it was a way to motivate users to participate at the *LightZone* forums. But even without any registration it's possible to view annotated screenshots, watch videos and consult tutorials in PDF from the *Lightzone* website.

Of course, for novice users the most useful materials are those step-by-step guides where you can instantly see which settings lead to certain results. It's particularly surprising given the fact that *Lightzone* has a dramatically smaller user base than *Gimp*. You can find quickly help for *Gimp* online, including videos recorded and shared on YouTube by enthusiasts.

Both *Rawtherapee* and *Darktable* have superb online documentation of all sorts, including freely available PDF books (Rawpedia and *Digital Photo Development with Darktable,* respectively), and it's hard to tell which community is larger. The problem, though, is that a RAW file isn't easy to edit, and both *Rawtherapee* and *Darktable* gives you a hundred ways to make it worse. For instance, while the noise reduction tools are incredible, in the wrong hands they end up fighting each other. We recommend that you consult a handbook before doing anything.

That sort of things can be a challenge when working with *Gimp*. On one hand, there are numerous tutorials and baby steps for carrying out simple editing in *Gimp*, but high-profile tasks like denoising are only covered in a range of short articles, and require you to conduct trial-and-error experiments. That's why we



RawPedia (http://rawpedia.rawtherapee.com) is a must-read site for every photo enthusiast who's keen to improve their knowledge of the RAW format.

recommend splitting tasks between *Gimp* and a RAW editor (with which *Gimp* can possibly integrate).

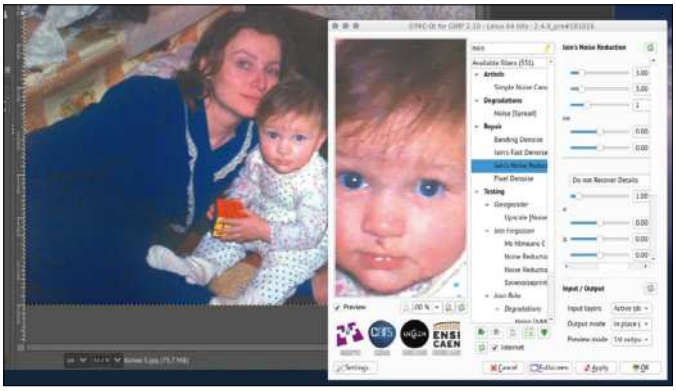
Predictably, *Pixeluvo* has less to offer, although there are still some useful sources of help. As a commercial product, *Pixeluvo* simply has to offer some support, including training materials and documents. The available amount is limited and mainly lives at the main web site (**pixeluvo.com**). We can't blame the developer for that because *Pixeluvo* isn't a popular product. The official *Pixeluvo* forum isn't very lively either, though...

VERDICT

GIMP	9/10	LIGHTZONE	8/10
DARKTABLE	10/10	PIXELUVO	7/10
RAWTHERAPEE	10/10		

Darktable and Rawtherapee come top for online help sources.

Photo editors
The Verdict



The winner of this month's *Roundup* was easy to identify. *GNU Image Manipulation Program*, or *Gimp*, is an undisputed champion and the best choice for photographers, regardless of their workflow. *Gimp* is the best for painting and retouching, but it's also the most reasonable choice for repairing, retouching and applying various recovery techniques thanks to third-party extensions from G'MIC and FX Foundry. If you need to process RAW images, *Gimp* can integrate with UFraw and *Rawtherapee* and use them as image sources. It's a versatile and comprehensive software package.

Deciding on the positions of the remaining programs was more challenging, because they all came very close to each other. Securing second place is *Rawtherapee*, with its superb colour correction tools and a stronger denoising algorithm. It's hard to tell if *Rawtherapee* is really better than *Darktable* – both are just different but comparable in terms of features. Still, in our tests *Darktable* was slower and didn't have a capable white balance fixer, which is why it lost a few points.

The surprising news is *Pixeluvo* coming in third place. This lesser-known commercial application is short on professional tools and lacks a number of sophisticated (and often overlapping) features that other RAW processors have. However, *Pixeluvo* is equipped with brilliant editing tools, useful colour adjustment features, quick effects, layers and even its own RAW import dialog. It's the most balanced editor that is (probably) worth the price.

This means that *Lightzone* brings up the rear, which doesn't mean that it's a bad program in itself. However, we were disappointed that *Lightzone* – which calls itself a professional-level RAW image processor – had such a poor noise removal tool. *Lightzone* still performed well when fixing colours and removing spots, but denoising could just be a little better.

So, for most use cases we recommend using *Gimp* and *Rawtherapee*, either separately, or by turning *Rawtherapee* into a RAW import plug-in for *Gimp*. If you don't need to reduce noise and grain, have a look at *Pixeluvo*, which has everything for amateur photo processing. Don't ditch *Lightzone*, either: it has certain advantages as a good second-tier photo-editing program.

1st Gimp 10/10

Web: **www.gimp.org** **Licence:** **GPLv3**
Version: **2.10.8**
The one software package for all your photographic needs. Impressive!

2nd Rawtherapee 9/10

Web: **https://rawtherapee.com** **Licence:** **GPLv3**
Version: **5.5**
It has more features than necessary, but it processes RAW shots perfectly.

3rd Pixeluvo 8/10

Web: **www.pixeluvo.com** **Licence:** **EULA**
Version: **1.6**
It's powerful and easy to use, but is still a lighter weight class of software.

4th Darktable 8/10

Web: **www.darktable.org** **Licence:** **GPLv3**
Version: **2.6**
It has masks, profiles and a great denoiser, but it falls a bit short in other areas.

5th Lightzone 6/10

Web: **www.lightzoneproject.org** **Licence:** **BSD**
Version: **4.1.9**
Capable image-editing software, but with poor grain/noise removal tools.

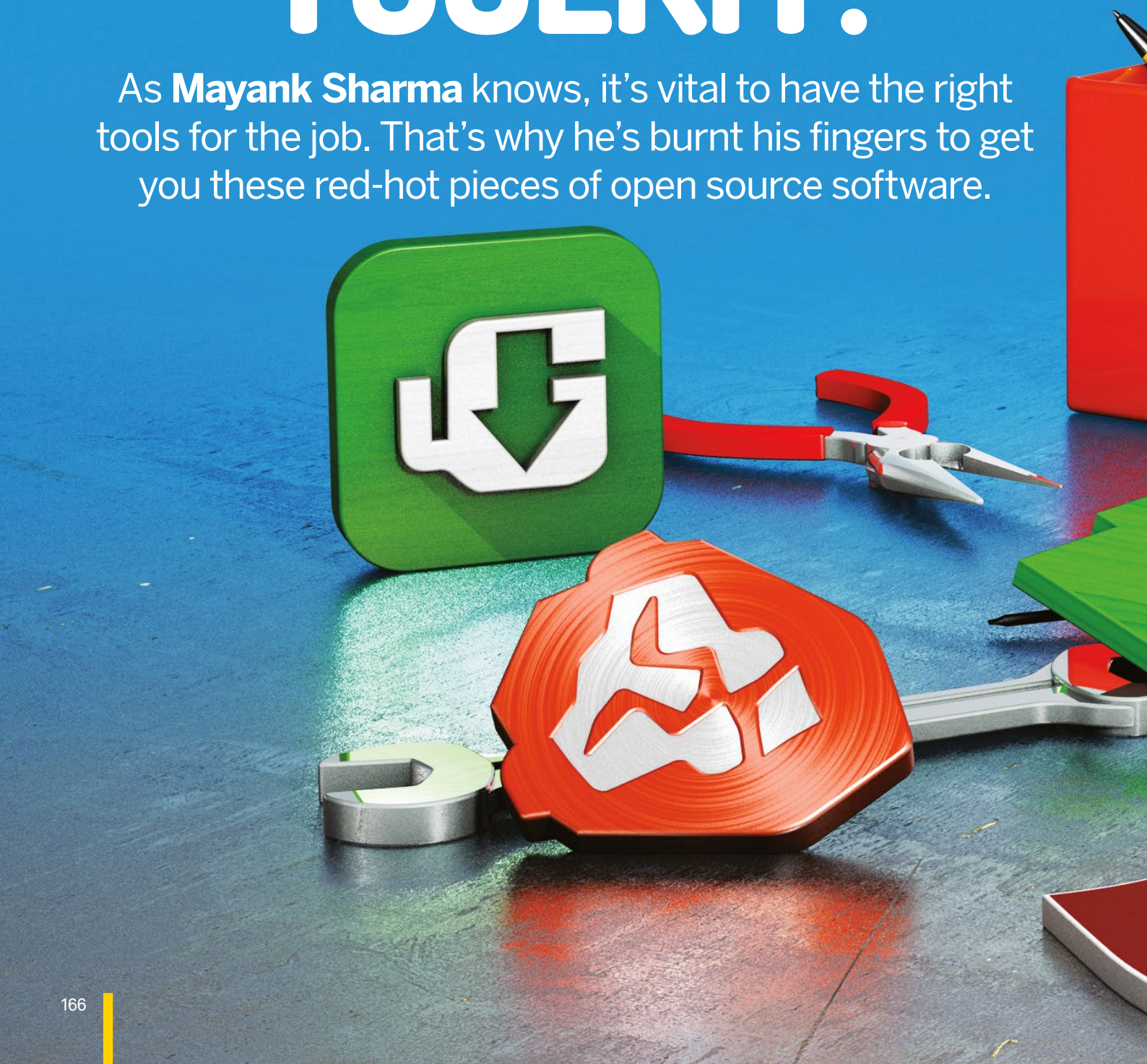
» ALSO_CONSIDER

Some people will argue that we forgot about *AfterShot Pro* from Corel, as well as some heavy-weight open source packages like *Digikam* or *Krita*. Well, *AfterShot Pro* is proprietary software with a price tag to boot, and we certainly don't want to turn our group test into an OSS vs proprietary software war.

Digikam is very good indeed and delightful to work with, but it focuses on collecting and sorting photos rather than on editing and processing. Finally, *Krita* is a great tool for digital artists, comics writers and artwork designers, but it's far less robust when used for editing photographs. Other alternatives also exist, and you may have read about them in previous issues of *Linux Format* (for example, *Photivo (see* **LXF238***)* and *Fotoxx*, to name just two). We, of course, acknowledge these programs and other free software packages for editing photos, but for this *Roundup* we selected the most powerful and feature-rich titles that can be used on a daily basis. LXF

THE ULTIMATE OPEN-SOURCE TOOLKIT!

As **Mayank Sharma** knows, it's vital to have the right tools for the job. That's why he's burnt his fingers to get you these red-hot pieces of open source software.



The mainstream Linux distributions spend a lot of time selecting their default bouquet of programs. After all, they need the tools to cater to the largest number of users. Sure, these programs excel at what they do, but there are thousands of other brilliant software lurking in code-sharing silos like GitHub and Sourceforge, and even inside the official repositories of your favourite distribution, waiting to be discovered.

Realistically though, do you have time to try all of these tools? Do you even want to, considering that your needs are served well enough by the distro's default options? Add to that the fact that most of us have our favourite open source apps and a conviction that they work for us better than any available alternative. That said, we'll encourage you to take a step outside your comfort zone to marvel at the diversity of the Linux-verse. And don't let the number of low-quality, unmaintained programs deter you from exploration.

EXPAND YOUR FRONTIERS

"Don't let the number of low-quality, unmaintained programs deter you from exploration"

At *LXF Towers* we spend a lot of time rummaging around the source code mirrors and other places to discover new gems to cover in the magazine. This issue we've decided to collate all the best ones we've encountered in our travels. We've stayed away from including the popular mainstream tools, but you might still be familiar with some of them because they're just that good.

For the most part though, the next few pages will introduce you to graphical apps and CLI utilities that you've never come across before. No matter the type of user you are or how you use Linux, the following pages will give you plenty of open source goodies to spruce up your Linux box.

Image tools

Check out these nifty options for managing or editing your image collection.

PENCIL 2D

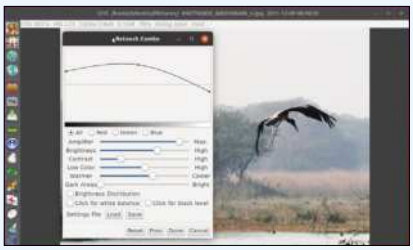
You can use *Pencil 2D* to create both static and animated drawings. It's a comprehensive program that will help digital designers create professional-grade cartoons and animations in multiple layers. The intuitive tool is well complemented by the video tutorials on its website. **www.pencil2d.org**

DARKTABLE

If you work with RAW images, *darktable* will help you create a professional digital darkroom to process them. The tool's interface gives you access to a full gamut of image manipulation and editing tools on par with commercial heavyweights from the likes of Adobe and Apple. **www.darktable.org**

FOTOXX

A comprehensive image-manipulation program with a rich set of retouch and edit functions that go beyond changing brightness, contrast and colour. Like most dedicated image editors, *Fotoxx* enables you to select an object or area within an image using various tools such as freehand outline, follow edges and select matching tones. Another unique feature of the tool is that it makes it possible to edit the images without using layers. You can also use *Fotoxx* to create HDR and panoramic images, reduce noise and remove dust spots, create collages, mashups, and slideshows with animations. There are also several artistic effects to help convert a photo into a line drawing, sketch, painting, embossing, cartoon, dot image, or mosaic.



You can use Fotoxx on resource-strapped machines and the program can be navigated entirely with the keyboard.

On the initial launch, *Fotoxx* fires up its quick start guide in the web browser, along with a dialog to index your image library. This process can take some time depending on the number of images you have in your library. All its features can be accessed from the relevant options that are clearly labelled in the left-side panel. **https://kornelix.net/fotoxx/fotoxx.html**

RAPID PHOTO DOWNLOADER

This program might seem redundant considering that most photo-management tools can import photos themselves. *Rapid Photo Downloader*, however, is designed for transferring photos and videos. It offers a lot more functionality that make it a perfect tool for downloading, processing and organising photos and videos.

The tool gives you control over how it processes and sorts the downloaded photos. The program's default rules automatically transfer the downloaded photos inside date-based subfolders. You can also define custom rules as per your requirements. For example, you can ask the program to sort photos by their type, which comes in handy if you shoot both RAW and JPEGs. In fact, you can create complex subfolder hierarchy by defining naming rules based on Exif values, such as focal length, ISO, and so on. You can also specify an external USB storage device as the back-up destination, and the program will automatically back up the photos while downloading them from the camera. **https://damonlynch.net/rapid**

CONVERSEEN

Converseen is a straightforward frontend to various command-line conversion utilities that can convert images to and from over 100 formats, rotate and flip them, change their dimensions, and rename them in a fraction of the time it would take to perform these tasks manually. **http://converseen.fasterland.net**

Manipulate any number of images within the Action panel.

FROGR

Use this tool to upload content to Flickr and also access its basic upload features such as the ability to describe images, set specific licenses and categorise them into sets and group pools. **https://mariospr.org/category/frogr**

NOMACS

This image viewer can display images in all the popular image formats. In addition to the usual features it has some surprising ones such as the ability to synchronise viewing between multiple network instances. **https://nomacs.org**

ENTANGLE

Use this tool to tether and control your camera from the computer. *Entangle* will also help you set up the shot by tweaking the aperture, shutter speed, ISO and other settings and then download the image. **https://entangle-photo.org**

FLAMESHOT

You can use *Flameshot* to capture the whole or a specific portion of the screen. Additionally, it also provides you with a whole set of drawing tools to add annotations to your screenshot. **https://github.com/lupoDharkael/flameshot**

LYCHEE

Lychee has a simple user interface, and is a platform for storing and sharing photos, You can run it from a SBC like the Raspberry Pi and use it to upload, manage and share photos. **https://lychee.electerious.com**

Audio and Video

Edit your videos within an inch of their lives, and organise your music.

UMS

This streaming server will transform your Linux desktop into the ultimate media streaming station to broadcast media to other computers, smartphones, tablets, gaming consoles and even TVs. *UMS* streams media via the Universal Plug and Play (UPnP) protocols to any DLNA-compliant device. You'll have to roll-out *UMS*, although the procedure is fairly simple.

The first time you launch it, *UMS* takes you through a basic three-step configuration wizard. Although you can start streaming without further configuration, UMS does include an admin panel that offers customisable options and tips to guide new users. It also includes a minimal web interface for streaming your content.
www.universalmediaserver.com

MPV

A resource-conscious video player that also looks good on modern machines. Based on *mplayer2*, *MPV* continues the tradition of *CLI* by introducing optimised and cleaned-up code with new configuration options and features. It offers a minimal user interface that stays out of the way, enabling you to watch your videos in peace.
https://mpv.io

SHOTCUT

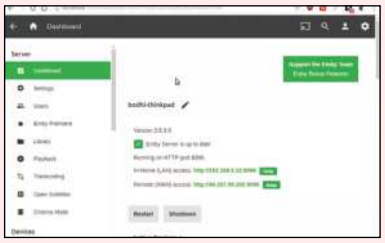
A fairly advanced video editor, *Shotcut* is one of the few tools that supports editing 4K videos. The program supports many audio and video formats along with a variety of transitions and effects. It has a long list of impressive features and a collection of video tutorials to help orient first-time users.
https://shotcut.org

CURLEW

Powered by the *ffmpeg* library, *Curlew* is an easy-to-use media converter. It has the ability to crop and pad videos and convert only specified portions of files. Select from one of its preset settings for various devices and formats and you'll also be able to preview the files before the conversion.
https://curlew.sourceforge.io

EMBY

This is an all-inclusive media server that comes with plenty of features. *Emby* is a one-stop shop for accessing and viewing all your media on any device: from other computers and mobile platforms to Android TV, Chromecast, Roku, Xbox and more. Besides local folders, *Emby* enables you to add network shares, which is a great time-saver if you have media on different computers in your network. The *Emby* server is available as a pre-packaged binary and its website has installation instructions for all popular distros including Arch, CentOS, Fedora and Ubuntu. You can right-click items to reveal useful options such as downloading them, adding them to playlists and collections, converting them or editing their metadata. You can change the stream quality during playback and display full-screen. *Emby* also enables you to set up parental controls, and you can block items that have a higher rating than the maximum you've defined.
https://emby.media



Emby can also connect to TV tuner cards and DVR devices to stream and record live TV.

AIRSONIC (35)

Based on the now closed-source project, the *Airsonic* audio streamer makes your music omnipresent. It does on-the-fly conversion and can stream files in almost any format to multiple players simultaneously.
https://airsonic.github.io

TUPITUBE

A set of tools to encourage kids to create 2D animations, *TupiTube* has an easy-to-use interface with just the right number of features and a host of textual and video documentation to get started.
www.maefloresta.com

PICARD

A nifty little tool, *Picard* will get your music collection back into shape. The program can sort your music library and fill in missing tags, rename oddly named files and identify incomplete albums. **https://picard.musicbrainz.org**

TRAVERSO DAW

A powerful audio recording and editing platform, *Traverso* is a digital audio workstation tool that can do everything from creating simple recordings to editing multi-track audio. **https://traverso-daw.org**

MIXXX

If you need to play DJ at an event, run through your music library with *mixxx* and use its scratchable waveform to loop beets, alter the tempo of tracks and change their pitch to create your own mix.
www.mixxx.org

Disk tools

Keep your hard drive in tip-top shape, recover lost data or delete it safely.

FOG

The constant barrage of repetitive tasks such as running checks, troubleshooting errors, swapping out dead hard disks, doing fresh installations over and over again can sap the energy out of any system admin, irrespective of the size of their network. Using *FOG* you can image and clone machines from the comforts of the admin HQ.

FOG is a complex piece of software and offers plenty of options, with various fields to describe the host images. It can also arrange them into groups for easier management. There are also several options to schedule the imaging process.

FOG can also be used to debug imaged computers, remotely wipe hosts and more. The server also handles regular admin tasks such as installing software and can even manage printers on the network. The *FOG* server is scalable and can manage large networks spread over multiple locations in the same building or on the other side of the planet. One of the most useful features of the *FOG* server, especially for admins of larger networks, is the multicast ability. Using this feature you can deploy multiple machines in one go. To supplement it on such large networks, you can have multiple *FOG* installations configured as storage servers that help take the load of the main *FOG* server when imaging computers.

https://fogproject.org

DUPLICATI

An easy-to-use back-up application, *Duplicati* enables you to fine-tune the list of locations you want to preserve by either defining filters or toggling one of the predefined options to exclude certain types of files. You can manage the tool via a browser-based interface. The program breaks down critical tasks into wizards and also exposes just the right number of features for the job in hand, while advanced options are just a pull-down menu away.

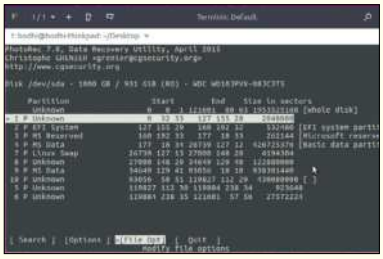
Files are encrypted with AES-256 and you can also use GPG before sending the backups to their destination. *Duplicati* also compresses all data before it's encrypted. It supports Zip and 7Z compression, and can skip compression of already compressed files such as MP3 and JPG.

www.duplicati.com

PHOTOREC

Photorec is a nifty little command-line based tool that can restore accidentally deleted files. When you delete a file, the file system just marks it as deleted, and makes the space the file occupies available to other files. *Photorec* works by recovering such files that are missing regular metadata such as a filename. Despite its name, the CLI utility can sniff files in various formats.

www.cgsecurity.org/wiki/PhotoRec



Photorec is part of almost every recovery distro and it ships along with TestDisk.

GPARTED

One of the best graphical utilities to manage partitions, you can use *Gparted* to create, delete, resize, move and copy partitions while preserving the data they house. The program works with all kinds of hard disks, SSDs, and RAID devices and supports all the filesystem in vogue.

https://gparted.org

SECURE-DELETE

The secure-delete package contains various utilities to securely delete files and wipe all traces of data in the free space on the disk. There's *srm* that make it impossible to remove any deleted files, *sfill* to wipe all data from the free space on the disk, and *sswap* to wipe data from the swap partition.

http://srm.sourceforge.net

CDEMU

The *CDemu* tool enables you to mount disc images in various formats including .ISO, .bin and .nrg. The *CLI* tool also has several graphical front-ends for Gnome and KDE desktops.

https://cdemu.sourceforge.io

DUC

If you're looking to free up some disk space, use *Duc* in the first instance to gain a better idea of the files occupying the space. The tool does a better job in inspecting your hard disk than the file manager.

https://duc.zevv.nl

DDRESCUE

Try image a failing drive with *ddrescue* before attempting recovery. The utility doesn't write zeros to bad sectors, and tries to fill in the gaps without wiping out the data already rescued.

www.gnu.org/software/ddrescue

PYDF

A replacement for *df*, *pydf* is a Python script that highlights the different types of filesystems. It has an large set of configurable parameters and various options to control its output.

https://pypi.org/project/pydf

SMARTCTL

Bundled as part of the *smartmontools* package, the *smartctl* utility controls and monitors the SMART system built into hard disks. Use the tool periodically to run self-assessment health checks on the disks.

www.smartmontools.org

File management

Control your data, where it can be accessed, and archive it, too.

EICIEL (45)

Everything in Linux is a file and the access control list (ACL) helps determine the access rights for each file. You can modify the permissions from the command line or use the graphical *Eiciel* utility that's in the official repositories of most distros.
https://rofi.roger-ferrer.org/eiciel

FSLINT

You can use the graphical *FSLint* tool for a comprehensive cleanup of your file system. It can remove duplicate files, temporary files as well as files that are otherwise difficult to locate and remove, such as files with invalid names, empty directories and bad IDs.
www.pixelbeat.org/fslint

NEXTCLOUD

Online storage services are a convenient option for accessing and sharing your data anywhere on the planet. With *Nextcloud* you can roll out a hosted storage server that gets you the convenience of an omnipresent storage service without shelling out wads of cash and your data to a third party.

Installing and rolling out *Nextcloud* is a fairly straightforward and well documented process. It has a feature-rich and intuitive web interface that shouldn't prove to be difficult to navigate even for first-time users, which is a feat in itself considering the sheer number of features it wraps underneath. The project scales well and can be used on a home network as well as an enterprise one. You can use its administration section to hook it up with other related network services such as a directory server to import users.

Sharing files with other users on the network or publicly via URLs is simple enough. The server ensures that changes made to shared files are synced to all users. It can also access files stored on a variety of cloud services such as Amazon, Google and Dropbox.
https://nextcloud.com

Nextcloud has an extensive plugins framework that enable you to use it for many deployments.

OPEN MEDIA VAULT

If you need more protection for your data than a simple back-up solution, then you need to convert an unused computer (or even a Raspberry Pi) into a dedicated network attached storage (NAS) device with the Debian-based *Open Media Vault* (*OMV*) server.

OMV is straightforward to roll out and simple to manage, thanks to its well-designed browser-based user interface, which makes it suitable for even non-technical users. It supports all the popular deployment mechanisms, including several levels of software RAID, and you can access the data it holds using all the popular network protocols such as SSH, SMB/CIFS, FTP and Rsync. The server also has an extensive permissions systems to control access to the shared volumes and folders. The server is modular and can be extended with a variety of official and third-party plugins. For instance you can turn your NAS into a torrent client to download data directly into the NAS storage or use it to stream stored music across the network.
www.openmediavault.org

SYNCTHING

The open source alternative to the popular but proprietary *BitTorrent Sync*, *Syncthing* can sync your files between computers over your LAN or across the web. The service uses a global discovery server to connect clients anywhere on the Internet.
https://syncthing.net

Use Syncthing's browser-based interface to add shared folders and all of your various devices.

CATFISH

A graphical alternative to the find and locate CLI utilities, *Catfish* is a versatile tool that has various options to help you prune the search results and find the file you were looking for.
www.twotoasts.de/index.php/catfish

PEAZIP

The graphical compression utility can read most archiving formats and boasts of some useful data security features such as the ability to create archives with encryption and two-factor authentication.
www.peazip.org

EXIFTOOL

A CLI utility for working with metadata in images, *exiftool* helps you manipulate the files based on their metadata. Users can also use it to strip all metadata before sharing images online.
www.sno.phy.queensu.ca/~phil/exiftool/

TKDIFF

A graphical front-end to the *diff* utility, you can use *TkDiff* to compare two files. The tool can connect with a range of source code management systems and has several other features.
https://tkdiff.sourceforge.io

RANGER

Ranger has a curses-based interface that works inside the console. The tool supports *Vi* keybindings, makes use of the rifle file launcher and is able to preview images and videos.
https://github.com/ranger/ranger

Communication

Chat securely or set up your own social network. In your face, Facebook!

JITSI

An all-in-one instant messaging app, *Jitsi* has an impressive set of audio and video-conferencing features. As a VoIP client, it enables you to make calls using the Session Initiation Protocol (SIP). It has all the features you'd expected from a softphone in that it can mute, put on hold, transfer and record calls. It can also make registrar-less SIP calls to other *Jitsi* users on the local network.

You can use *Jitsi* to make video calls to one user or to several on both SIP and XMPP networks, while using *Jitsi Videobridge*, you can host multi-user video calls. It's also a capable IM client with several unique features including the ability to stream and share the desktop of users at both ends of the call at the same time.

https://jitsi.org

TOX

If regular IM clients aren't secure enough for you, use the *Tox* IM protocol which encrypts all chats with the NaCl encryption library and instead of a central server, routes them over direct P2P connections between users. The IM uses *Tox* IDs, which are public keys of peers instead of a user account and also allow for greater anonymity.

https://tox.chat

There are several apps that can communicate using the Tox protocol and one of the most popular ones is qTox, which works on several platforms.

BEEBEEP

An IM client with a difference, *BeeBEEP* is designed to connect you directly with peers on your network. It'll automatically detect other *BeeBEEP* users on the network and provides all the features of a modern chat client and apart from encrypted text messages, can also share files.

http://beebeep.sourceforge.net

OPENFIRE

There are several XMPP-based IM servers available but *Openfire* is one of the easiest to manage. It implements many of the commonly used functions of the XMPP protocol. While you can use any XMPP-compatible IM client to chat through *Openfire*, it works best with its own *Spark* client. **www.igniterealtime.org/projects/openfire**

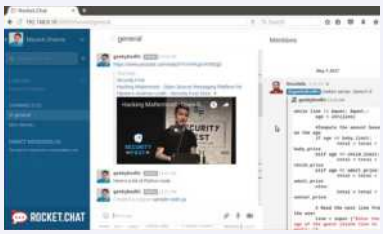
ROCKET.CHAT

Billing itself as the 'ultimate chat platform', *Rocket.Chat* is filled with useful features. In addition to essentials ones like threaded conversations and the ability to edit and delete messages, the collaborative communications server offers several interesting ones such as live chat, with which you can use *Rocket.Chat* to have conversations with visitors to your website.

You can roll out *Rocket.Chat* on a snap-supported distro with a single command, and it also support deployments over Docker and a wide range of PaaS services. It can also import your data from another service like *Slack* or *Hipchat* and with the SlackBridge plug-in can also mirrors messages received in a *Slack* channel or private group into *Rocket.Chat* in real-time.

Rocket.Chat also enables you to have a video conference with your team. You can also record and send voice messages to a public or a private conversation.

https://rocket.chat/community



Rocket.Chat enables you to preview links shared during a conversation including those from popular services like Twitter and YouTube.

SIGNAL

The *Signal* messaging app helps secure communications over the mobile network. The app is endorsed by top security experts and uses end-to-end encryption to secure all your text, audio and video calls.

https://signal.org

RIOT.IM

The Matrix protocol enables users registered with one service provider to seamlessly communicate with users of another. *Riot.im* is one of the best clients based on the federated Matrix protocol.

https://about.riot.im

RETROSHARE

A P2P communications and file sharing platform, *RetroShare* creates encrypted connections between friends. You can add friends by sharing your keys privately or you can exchange them via a chat server.

http://retroshare.net

SOGO

One of the best alternatives to MS Exchange, *SOGo* has all the necessary groupware functions with native desktop and mobile clients. Try the online demo before deploying it on your server.

https://sogo.nu

HUMHUB

With *HumHub* you can deploy your own social network on the internet network. It has all the standard social networking features you'd expect and the project's website hosts an online demo as well.

www.humhub.org

Productivity

Put some order into your life with these finance, web filtering and admin tools.

OSMO

A personal information manager (PIM) helps you keep track and organise all the information coming in from various sources. *Osmo* is a lightweight PIM that helps manage appointments, tasks, contacts and notes. The application has a straightforward and integrated interface with four tabs on the top for Calendar, Tasks, Contact and Notes. The program lacks any menus, but the action-linked buttons at the top change as you switch between the four components. The Calendar pane provides a simple functional calendar along with some related information such as the week number and the number of days to the end of the year. You can double-click a date to add notes or right-click a date to add a task.

The options for defining a task in *Osmo* are housed within a basic and an advanced tab. Use the basic tab to define the due date and time and assign the task a priority, while the advanced tab enables you to mark the task as recurrent and define relevant properties. You can store the address of your contact using the Contacts pane. It sports additional features including a search function, a birthday browser, as well as the ability to point out the address of a contact on Google Maps. The Notes pane makes it possible to jot down text using rich text editing functions and you can optionally encrypt all your notes using a password. **http://clayo.org/osmo**

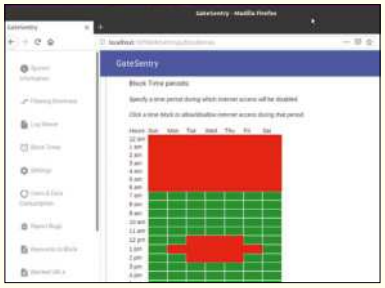
FOCUSWRITER

Scribble without distraction with the *FocusWriter* text editor, which runs full-screen and uses an auto-hide menu for a clean workspace. The editor supports the popular text formats and has basic support for ODT files as well.

The primary goal of *FocusWriter* is to enable you to focus on the writing, so while the tool might lack some of the dexterity of the mainstream text editors, it has several unique features that are more tuned towards this objective. You can set alarms to trigger after a certain period has elapsed, or at a particular time and also set yourself targets in *FocusWriter*. An interesting feature is Focused Text, which fades out everything except the section you're typing – a whole paragraph, a block of three lines, or just the current line. **https://gottcode.org/focuswriter**

GATESENTRY

A simple web filter, *GateSentry* works on Linux as well as the Raspberry Pi. It's easy to deploy and configure. *GateSentry* authenticates users and filters traffic by websites, content type as well as by time. The simple proxy server can display data consumption statistics and can also enables you to remotely disable a user's access. **http://gatesentryfilter.abdullahirfan.com**



GateSentry can also filter traffic over HTTPS once you install its certificate on your devices.

KIMAI V2

Kimai tracks work times and prints out a summary of your activities. It can be used for a single user or multiple ones and can manage multiple customers, projects. The program has a browser-based dashboard that you can fiddle around with using the demo installation on the project's homepage. **https://v2.kimai.org**

SIMPLESCREEN RECORDER

Contrary to its name, *SSR* is flush with features and gives its users a good amount of control over the screencast. The tool can record the entire screen and also enables you to select and record windows and regions on the desktop. **www.maartenbaert.be/simplescreenrecorder**

TURTL

A note-taking tool with a focus on privacy, *Turtl* has an impressive list of features. It can create different types of notes and makes it possible for you to attach images and other files to the notes that you've created. **https://turtlapp.com**

DICTION

A CLI utility, *Diction* helps flags words that are commonly misspelt. It can create better copies by flagging words that are commonly associated with beginner's mistakes and to even suggest better wording. **www.gnu.org/software/diction**

TASK COACH

A simple to-do manager, *Task Coach* simplifies the process of defining and following up on tasks. You can also use the tool to assign a priority to tasks along with a completion date and an optional reminder. **www.taskcoach.org**

EQONOMIZE

Eqonomize is a personal expense manager with a clean interface and all the features you need to help organise your finances. The program can track single or multiple accounts and fix budgets. **https://eqonomize.github.io**

STRETCHLY

A simple app that pops in to remind you to take a break. It has a couple of preset break intervals and the break windows share ideas to help you relax. You can control the tool from its tray icon. **https://hovancik.net/stretchly**

System Administration

Find out exactly what's going on under the hood of your Linux box.

GLANCES

Glances is a Python script that smartly displays a lot of information about your current session inside a standard terminal window. You can use the tool to monitor remote machines and use the built-in web interface to monitor machines using a web browser.
https://nicolargo.github.io/glances

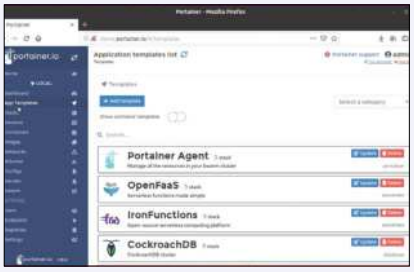
FIREJAIL

Firejail enhances security by isolating programs and processes inside limited sandboxed environments. By locking away the web browser or any tool that can be compromised, *firejail* prevents a compromised program from accessing critical areas of the filesystem.
https://firejail.wordpress.com

PORTAINER

Using Docker via the terminal isn't all that cumbersome and the tool is well documented. To make your life easier however, you can use *Portrainer*, which is an open source, web-based graphical front-end that supports all the features exposed by the Docker API. You can use it to manage containers, images, networks, and volumes and it can manage a standalone Docker environment, or a Docker Swarm.

From its dashboard you can create containers and view and manage all available ones. You can obtain real-time stats such as CPU and memory usage and various other details about the running containers and even access a container's console, all from within the browser window. *Portainer* is available as a Docker container so you can install it with a single command. To create containers you can use one of the dozens of predefined templates. The tool can be used by multiple users and has useful user-management functions. You can for example use it to define the levels of access any other users have to *Portainer*, and the aspects of Docker they can manage from within *Portainer*.
https://portainer.io



You can use Portainer to pull images from Docker Hub or a private registry.

BOOTCHART

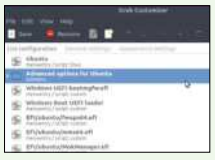
One of the major causes of longer boot times is that your system starts unnecessary tools and services during startup. But before you axe them, it's best to get a picture of what's happening while your distro boots up.

Bootchart is a simple utility that enables you to profile your Linux boot process and help measure the loading times of different services. It's now merged with Systemd. Fire up a terminal and enter `systemd-analyze time` to know the breakup of the boot duration. Similarly, `systemd-analyze blame` will list all running units ordered by the time they took to start.

To generate an image of the boot process, type `systemd-analyze plot > boot.svg`. From this image you can find all the active processes and can remove any you don't need. For example, if you print occasionally, you can disable CUPS from starting at the boot time. Furthermore, the image also helps you spot processes that take control of all resources and force the other processes to wait, slowing boot up.
www.bootchart.org

GRUB CUSTOMIZER

There are several reasons you might want to change the default boot loader behaviour and the *Grub Customizer* tool will help you do just that. The graphical tool can modify all aspects of the boot process and can even tweak how the Grub screen looks. **https://launchpad.net/grub-customizer**



Grub Customizer also helps restore order when you've accidentally wiped clean the MBR.

REMMINA

A very usable remote desktop client, *Remmina* supports a wide range of protocols. It performs well and offers useful features such as the flexibility to change the quality settings of the connection on the fly.
https://remmina.org

INXI

A CLI tool that lists info about the hardware in your computer. With *inxi* you can collate details about the kit, including vendor details, device driver configuration and more.
https://github.com/smxi/inxi

POWERTOP

Intel developed the *PowerTOP* utility to track down apps that eat up your laptop's battery. The CLI utility offers suggestions to tweak power-guzzling tools to squeeze more juice from your laptop's battery.
https://01.org/powertop

STACER

Stacer is a system monitor that helps you track different aspects of your installation. It can also show alert messages when values for a parameter exceed a certain threshold.
https://oguzhaninan.github.io/Stacer-Web

LOGWATCH

Logwatch parses, analyses and filters logs and then generates daily reports on your system's log activity. The utility also enables you to control the level of the report's verbosity.
https://sourceforge.net/projects/logwatch

Internet tools

Make your online time more efficient with this collection of utilities.

UGET

By default, the lightweight *uGet* download manager relies on *curl*, but if you install the aria2 package, it can take on some more features, such as the ability to download torrents.

UGet is an all-round downloader that has all the features you'd expect from a download manager. It features a download queue, can pause and resume downloads and also accelerates downloads by grabbing files from multiple parallel streams. Furthermore, batch downloads are one of its specialities. You can use the tool to prioritise the download queue and even regulate the speed of the downloads individually. *UGet* can also shut down and hibernate your computer once it's finished downloading all the files.

https://ugetdm.com

BRAVE

While popular browsers can be equipped to thwart information leaks, *Brave* ships with privacy-strengthening features. The Brave project promises two improvements over the competition: privacy and speed. One of them is achieved by blocking ads and trackers, which has the pleasant side effect of improving the browsing speed.

https://brave.com

Since blocking ads may rob sites of revenue, Brave has some unique compensating tricks.

WALLABAG

A read-it-later tool, *wallabag* began when Google Reader was shuttered in 2013. You can install it on your own server or use it via its own service that costs about £8/year. *Wallabag* has impressive features including the ability to save articles, filter them by reading time, and more.

www.wallabag.org

TOR BUNDLE

Tor enables users to browse the web anonymously. Its goal is to prevent people from tracking you online. The *Tor Browser Bundle* includes everything you need to connect to the Tor network of relays including a customised version of *Firefox* known as the *Tor Browser*.

www.torproject.org/projects/torbrowser.html.en

RAMBOX

Most desktop distros ship with a default messaging tool that enables you to sign into multiple online messaging services. These tools can sign into only a handful of the popular services.

In contrast, *Rambox* can plug into over 100 online messaging and email services. Every added service resides within its own tab, from where you browse the message history, write messages to your contacts, or use the other means of communication supported by the service. The tool will also display notifications. *Rambox* can also sync configurations if you use it on multiple computers. The program respects your privacy. Instead of storing your personal data, *Rambox* uses the **partition:persist** attribute of the <webview> tag to create a persistent connection with the signed-in service. You can also set a master password and ask the tool to lock itself after a predefined period of inactivity.

http://rambox.pro

The tool is available as a Snap and AppImage binary that you can use without installing.

MAGIC WORMHOLE

A Python script that creates single-use encrypted channels to ferry files between computers across the Internet identified via pronounceable code.

https://github.com/warner/magic-wormhole

UBLOCK ORIGIN

The *uBlock Origin* browser extension is an ad-blocker that also blocks tracking servers, malware domains, and more. It's available in the app stores of all mainstream web browsers. **https://github.com/gorhill/uBlock/**

QUITERSS

QuiteRSS has all the features that you'd expect from a news reader. It can pull in RSS and Atom feeds, apply labels to each item and offers plenty of customisation options to boot.

https://quiterss.org

VOCAL

A feature-rich podcast grabber, *Vocal* can subscribe to and stream podcasts on-the-fly or download them for offline listening. It integrates with popular desktops and has smart library management functions.

https://vocalproject.net

UFTPD

uftpd ships with defaults that work for most users. It's designed for home users and developers who need a simple FTP server, but aren't too particular about being secure.

http://troglobit.com/projects/uftpd

Security tools

Ensure ne'er-do-wells don't get their hands on your system and files.

ZULUCRYPT

While you can control access to the data on your computer using user accounts and file permissions, they aren't enough to prevent a determined intruder from gaining access to your private files. The only reliable mechanism to keep your personal data to yourself is to encrypt it. Sure, working with encrypted data is an involved process, but it'll go a long way in reinforcing your security and insulating your data.

zuluCrypt is a graphical encryption tool that has an intuitive, easy-to-follow interface. Using the program you can create an encrypted disk within a file, a partition and even USB disks. It can also encrypt individual files with GPG. *zuluCrypt* can perform block device encryption, which means that it can encrypt everything written to a certain block device. The block device can be a whole disk, a partition or even a file mounted as a loopback device. With block device encryption, the user creates the file system on the block device, and the encryption layer transparently encrypts the data before writing it to the actual lower block device. While encrypted, the storage areas just appears like a large blob of random data and doesn't even reveal its directory structure. All these functions and more and accessed via the excellent user interface. **http://mhogomchungu.github.io/zuluCrypt**

GUFW

Gufw is the graphical front end for *UFW*, the uncomplicated firewall, which is one of the easiest front-end for *iptables*. The tool has a simple interface and includes predefined profiles to regulate incoming and outgoing traffic. You can alter the incoming and outgoing policies of the profiles as per your requirements. You can also use *Gufw* to define specific rules for allowing traffic for individual apps and services. **http://gufw.org**

Switch to the Report tab to get the live network traffic report.

BITWARDEN

The *BitWarden* password manager does end-to-end encryption and uses AES 256-bit as well as PBKDF2 to secure your data. It goes through this trouble because unlike its peers it stores your encrypted passwords in a remote Microsoft Azure cloud. The app can also import passwords from over 24 sources. **https://bitwarden.com**

OSQUERY

You can keep an eye on your network using the *osquery* CLI tool that enables you to query your devices just like a database. You can use the tool to check logged in users, keep an eye on the firewall, identify outdated kernels and keep an eye on the loaded kernel modules and the running processes. **https://osquery.io**

STEGOSUITE

Steganography is the art of concealing data inside another seemingly harmless message or image. The most common mechanism for implementing it is by replacing unused data in regular computer files with bits of information that aren't visible when viewing the original piece of data. Steganography is mostly used to complement encryption.

Stegosuite is a graphical tool that can hide text messages as well as any type of file inside an image. Fire up the tool, select an image, point to the files you want to hide inside it and enter a secret key to burn it in the image. To get the secret files from the image load it once again and use the Extract button along with the secret key to grab the hidden files and message from the image. **https://stegosuite.org**

MTR

MTR is a simple CLI network diagnostic tool that combines the functionality of the *traceroute* and *ping* utilities. So along with the path of the packet it displays a wealth of other relevant information. **www.bitwizard.nl/mtr**

AIDE

You can use *AIDE* to help spot intrusions by checking the integrity of the files by comparing their properties such as permissions against a baseline database created on the initial run. **http://aide.sourceforge.net**

NIKTO2

A CLI web server scanner that hunts for potential problems like server misconfigurations and outdated versions and version-specific issues. It can also perform automated tests against security vulnerabilities. **https://cirt.net/Nikto2**

JOHN THE RIPPER

A password cracker, *John the Ripper* is used for exposing weak Unix passwords. It's a CLI tool but also has a graphical interface called Johnny that exposes its various command line options. **www.openwall.com/john**

MALTRAIL

MalTrail is a malicious-traffic detection system that uses publicly accessible blacklists, custom user-defined lists and more to detect and log any malicious traffic. **https://github.com/stamparm/maltrail**

Developer tools

Make your coding life easier with this collection of capable utilities.

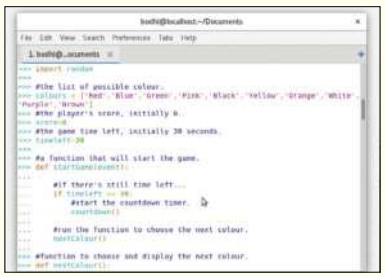
ATOM

Atom describes itself as a hackable text editor. Developed by GitHub, the tool has a built-in package manager that enables users to search and install plugins from within it. About 80 plugins ship with the utility by default. One of its more interesting features is the find and replace function that can also modify text across multiple files as you type.

Atom can also be used as an IDE and one of its highlights is the smart autocomplete feature. There's also a bracket matcher feature that highlights the line-number of the closing bracket corresponding to the one under your cursor. You can also define custom key bindings and add more functionality with packages for items such as minimaps and syntax-specific snippet libraries. **https://atom.io**.

BPYTHON

A Python shell that provides modern IDE-like features inside a terminal window. You can start using it with the default settings and then customise it by editing its config file. The shell does code completion and will also display a list of expected parameters. It highlights syntax as you type and the Rewind feature checks the entire code, which is kept in memory.
https://bpython-interpreter.org



Bpython enables you to save a session to a file, or even send it to pastebin.

ETHERPAD

Collaboration is key to any project. The *Etherpad* text editor enables users to collaborate on text in real-time. It runs inside the web browser and enables participants to interact via a text-based chat. It's full of features and you can take it for a spin on its website.
http://etherpad.org

VIRT-MANAGER

The *Virtual Machine Manager* is the open source graphical frontend for creating KVM-based VMs. It uses the *qemu-kvm* hypervisor, which is a version of the *qemu* machine emulator, and includes a *VNC* and SPICE client that displays a graphical console to the running VM. **https://virt-manager.org**

TURNKEY APPLIANCES

Installing network accessible software or web applications on a server can be quite a task because they require a lot of infrastructure software, which might take you hours to put together. From database servers to basic libraries, you'll have to spend quite a while to assemble a fully functional web server before you can deploy an application on the network.

Furthermore, if you plan to serve users from outside the network, you need to thoroughly examine your web server for any security leaks. This is a major undertaking in itself.

This is where *Turnkey Linux* shines. Using its virtual appliances you can deploy a new server app in no time. Put simply, a *TurnKey* virtual appliance is a self-contained system that packs in a fully functional instance of a web app with just enough components of an operating system to power that tool. The system works straight out of the box and can be deployed on either bare metal or on top of virtual hardware.

Turnkey Linux appliances are available in several formats depending on the hardware you want to deploy them on, from bare metal to OpenStack clouds. Importantly though, once they're up and running, irrespective of the platform, they all give you the same interface to administer and manage the web app.
www.turnkeylinux.org

OCELOT GUI

The database client can connect and interact with a MySQL or MariaDB server. You can use to create queries and it can highlight syntax and can fetch and display results from the **database.**
https://github.com/ocelot-inc/ocelotgui

DOXYGEN

Use this CLI tool to generate HTML documentation from the comments in your code. The tool supports many programming languages and can cross-reference the documentation with the code for referrals.
www.doxygen.nl

INFER

If you work with Java, C, C++ or Objective C, use *Infer* to weed out bugs The tool includes various analysis and its most interesting feature is that it can work across several procedures across various files.
https://fbinfer.com

MARK TEXT

The markdown editor supports the GitHub markdown spec and the CommonMark spec. It has a live preview, several editing modes and inline Math support and can export documents.
https://marktext.github.io/website

MANTA

When you've completed a project, use *Manta* to generate invoices and receipts. The invoicing app has a clean interface and several templates that you can customise as per your requirements.
https://github.com/hql287/Manta

SUBSCRIBE & SAVE UP TO 61%

Delivered direct to your door or straight to your device

Choose from over 80 magazines and make great savings off the store price!

Binders, books and back issues also available

Simply visit www.magazinesdirect.com

No hidden costs | Shipping included in all prices | We deliver to over 100 countries | Secure online payment

FUTURE

magazinesdirect.com

Official Magazine Subscription Store

Linux & Open Source Annual 2022

Everything you need to master open source software and distros

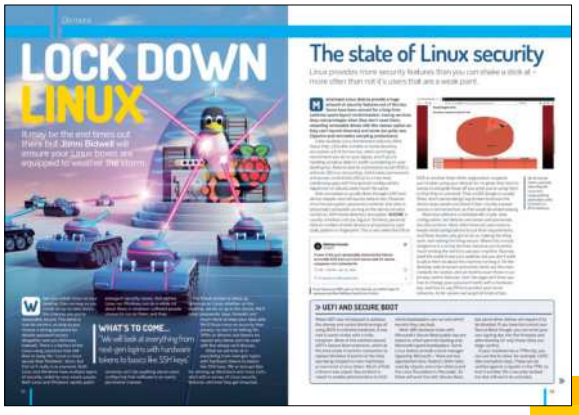
The best distros

Our pick of the best Linux distros

Do more with Linux

Record audio, build games and much more

Stay secure



Protect your private data and secure your system

FOSS tools

All the open source software you need to do whatever you want with Linux

BOOKAZINE