

LINUX

FORMAT

Get into Linux today!

60-MINUTE LINUX CRASH COURSE!

- » Easy step-by-step install
- » Customise the desktop
- » Games, video and more!

HOW TO HACK YOUR ROUTER

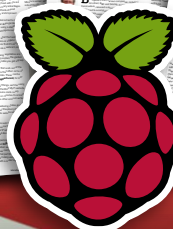
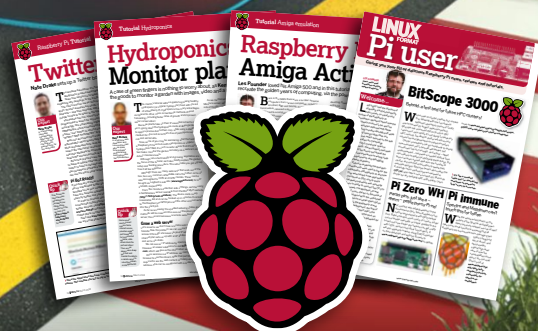
» Secure your Wi-Fi and unlock its full power



Coding Academy: Using Cmake like a pro developer



PLUS!
Meltdown & Spectre: how to protect your PC!



Sound & virtualisation

“Before the WebKit fork we were one of the largest contributors, behind Apple and Google”

Alberto Garcia on QEMU and Spanish open source



Plus: Pi User

Essential Raspberry Pi companion

- » Revive and run your Amiga games
- » Monitor and water your garden
- » Build an AI-driven Twitter bot

Software Snaps & Flatpaks

» Making open source easier and faster to install and use



Security Protect with Prey

» Using open source to lock and track devices





Toast Design is a
Heart Internet customer

Managed VPS

You take care of your
business. We'll take
care of your server.

Powered by

Heart Internet

All of the power. None of the hassle.

From **£39.99** per month



Fully monitored by
our experts



All updates installed
by us



Automatic site and
server backups



Plesk Onyx
control panel



WordPress Toolkit



Let's Encrypt included

Call us today about moving your sites for free

0330 660 0255

Heart Internet

Visit: www.heartinternet.uk
Email: sales@heartinternet.uk

What we do

» We support the open source community by providing a resource of information, and a forum for debate.

» We help all readers get more from Linux with our tutorials section – we've something for everyone!

» We license all the source code we print in our tutorials section under the GNU GPL v3.

» We give you the most accurate, unbiased and up-to-date information on all things Linux.



Who we are

This issue we asked our experts: We're introducing new Linux users to the fun and merriment of open source world, so what area/project/job do you enjoy the most?



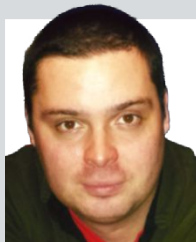
Jonni Bidwell

It's no fun when things go wrong, but figuring out what's gone wrong, why it's gone wrong and how you can fix it are all pretty enjoyable activities – once you start making progress anyway. Back when I used Gentoo, I think that I spent more time fixing Linux than actually using it. Hmmm.



Nate Drake

One day while skulking around my office I met an ethical hacker who, in exchange for an Oreo biscuit, introduced me to InfoSec. On reflection this was a fair trade as I was able to monetise my rampant paranoia and gain a fuzzy sensation from keeping others safe into the bargain.



John Knight

For me it's the desktop. It's my desktop. I'm a KDE guy myself, but no matter what you use, a lot of customisation is usually available. Features like virtual desktops and session management allow a bespoke environment bent completely to your whims that the proprietary offerings could never match.



Les Pounder

The most enjoyable aspect of open source is that it's all-encompassing. I no longer think of open source as just software. We now have open source hardware, for example Arduino and 3D printers. These devices are then used to create further open source projects that go on to spawn more and more!



Mayank Sharma

When I don my tech journalist hat, interacting with the project developers and users is one of the best aspects of the job. Unlike the corp comm types, the palpable passion of the entire ecosystem around an open source project makes the interactions more meaningful and lively.



Linux in 60 minutes!

» There's never been a better time to start using Linux and you've never had a better chance than with this month's issue of *Linux Format*! We're packing the ideal Linux starter pack with a bootable Live Disc that you can just insert and run, alongside a 9-page guide on getting up and running with Linux in just 60 minutes! Amazing.

We're standing on the shoulders of giants here. Dedicated, diligent, development geniuses have poured billions (probably) of hours into creating an open source ecosystem, which delivers an operating system kernel that's capable of powering super computers, world-spanning enterprises, your home desktop and the meek Raspberry Pi.

This flexibility and open nature means people can create beginner-friendly versions of Linux distros (that's what we call complete operating systems around these parts) with modern desktop interfaces and selected custom application suites. All this goodness comes wrapped up in a simple installer system to help get it onto your PC.

Linux distros don't bug you for updates, they don't snoop on you, there's almost no Linux malware, there's no bündelware and you're not locked out from playing, exploring and hacking the OS to your heart's content. If you want to have fun with your computers again, give it a try – you might like it!

The rest of the magazine will give you a taster of the huge expanse of areas the Linux kernel controls, and open source in general. From the exciting world of making and learning with the Raspberry Pi, to coding Python and creating virtual machines with VirtualBox. There's so much to include every issue that we simply don't have enough pages. Enjoy!

Neil Mohr Editor

» neil.mohr@futurenet.com

Subscribe & save!
On digital and print, see p30

Contents

Answers
is back!
On p14

"If our principles are right, why should we be cowards?" – Lucretia Mott

Reviews

AMD Ryzen 3 1200..... 17

A truly budget processor that delivers on the promise of a low price and quad-core performance that you can depend on. It might even be free of annoying flaws!



» It's a lean, mean processing machine, which is handy, as it's a processor!

Intel Core i3 8350K..... 18

What costs almost as much as a Core i5 processor but isn't as fast? That's right – a budget processor from Intel that, as it turns out, isn't really that low cost.

Freespire 3.0 19

We asked **Shashank Sharma** to judge how successful the revival of Lindows OS has been, which died back in 2008. Is the distro worth exploring in its current form?

Siduction 2018.01 20

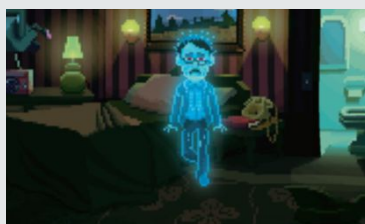
Shashank Sharma's interest is piqued by the sound of a distro based on apparently unstable software. After testing, he discovers the reality is a little different...

Daphile 17.09 21

Do you own plenty of music, but are struggling to set up a headless music server? Then **Shashank Sharma** might have a solution in the form of this distro...

Thimbleweed Park 23

Andy Kelly is hooked on this retro point and click adventure, and it's not that he's filling up on the delicious verb salad interface – he loves the murder mystery storyline!



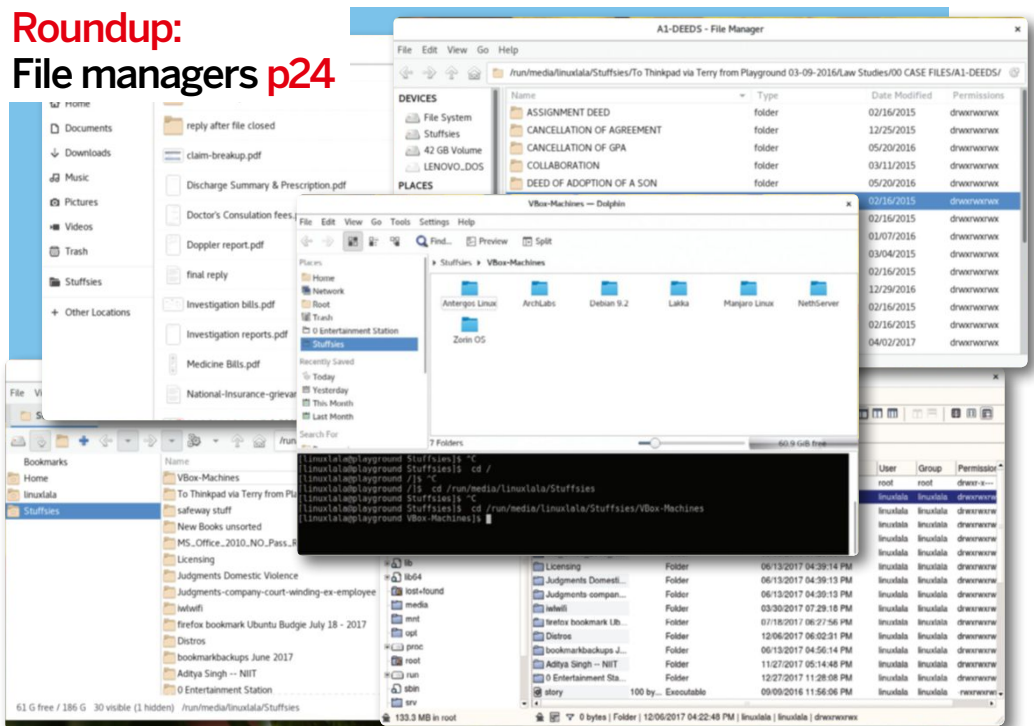
» Put that lawsuit away – Mulder and Scully don't want any trouble.

60-MINUTE CRASH COURSE!

Discover the most powerful operating system in the world – we take you from zero to hero in just 60 minutes. Get started on page 32!



Roundup: File managers p24



Interview



“I think I’ve still got the version of Debian that I installed back in 1997!”

Alberto Garcia on QEMU and Spanish FLOSS p42

GET STARTED IN LINUX Insert disc... Boot PC... Use Linux!

Elementary OS

A beautiful and easy-to-use Linux

- » Quick to install
- » Secure and safe
- » Every app you need

64-bit

BODHI LINUX Ideal for all older PCs

32-bit

On your FREE DVD

Elementary OS 0.41 64-bit

Bodhi Linux 4.4 32-bit

ROSA Linux R10 64-bit **p96**

» Only the best distros every month



Subscribe & save! 30

Raspberry Pi User

Pi news..... 52

What do you get when you mount 750 Pi 3s in a cabinet, discover what the new pins achieve in the Pi Zero, and more Spectre/Meltdown news.

Vivaldi web browser..... 53

Les Pounder loves stats and has found a way to monitor how many times he searches for pies.

Universal Amiga Emulator ... 54

The Amiga is close to **Les Pounder's** heart, and here he recreates the golden age of computing.

Twitter bots..... 56

Set up your own Twitter bot on the Raspberry Pi to spam the net, and then blame **Nate Drake**.

Garden Pi..... 58

Kent Elchuk proves his green-finger credentials are enhanced with a Pi monitoring system.



In-depth...

Snaps vs Flatpaks..... 46

People demand faster, easier distribution systems: **Mayank Sharma** checks out the next-generation of packaging and asks: is it for us?.



» Not quite the delicious Flatjacks we hoped for, but it's still pretty tasty.

Tutorials

Terminal MEncoder..... 62

John Knight tries to join the cool kids by ripping videos from the terminal with the big-brother of *Mplayer*.

Security Prey..... 64

Mayank Sharma has a Gollum-esque attachment to his devices and a Nazgul-like defence force on constant alert.

Encryption Entropy..... 66

Nate Drake guides you through the chaotic topic of increasing randomness to secure your system. Don't forget your lava lamp!

Virtualisation VirtualBox networks..... 70

Never happy with the default setup, **Mayank Sharma** shows you how to piece together a network, knowing that the firewall has all the pipes blocked.

Sysadmin Administeria..... 72

Dr Valentine Sinitsyn wonders about clouds, Amazon switching to KVM and then explains how to use *OSTree* – described as Git for operating system filesystems.

Routers Linux Embedded DE..... 76

Tired of proprietary firmware, **John Lane** gets to grips with LEDE (think an updated OpenWRT) and discovers that his router is actually quite good...

Coding Academy

Use CMake like a pro..... 88

Mihalis Tsoukalos teaches you about *CMake*, its configuration files and how to visualise project dependencies using *Graphviz*.

Array for Python!..... 92

Nate Drake guides you through the most fun you can have with Python while still sober.

```
1 cmake_minimum_required(VERSION 3.0)
2
3 PROJECT( simple )
4
5 set(CMAKE_BINARY_DIR ${CMAKE_SOURCE_DIR}/bin
6 set(EXECUTABLE_OUTPUT_PATH ${CMAKE_BINARY_DIR}
7
8 # For including the header file
9 include_directories(myLibrary/include)
```

Regulars at a glance

News..... 6

ALL THE PROCESSORS ARE BROKEN, but try and stay calm, the fix only slows I/O to a crawl and other bits of good news for 2018.

Mailserver..... 10

Readers' thoughts on computing paradigm shifts, thoughts on mistakes and thoughts on the USA.

User groups..... 13

Les Pounder on the virtues of paying it back with interest.

Roundup..... 24

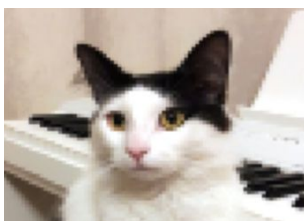
Shashank Sharma's life is too hard to handle, but at least he can sort out his desktop with the best file managers.

Subscriptions..... 30

We missed our targets so are being mercilessly treated in the *LXF* subs dungeon. Save us and subscribe!

Back issues..... 68

After the excesses of the New Year, lose weight with our guide to light and fast distros in *LXF233*.



» Our subscription team is waiting to take your call.

Overseas subs..... 69

We ship to every corner of the world, for a price, but what a deal – *LXF* goodness sent straight to your door!

HotPicks..... 81

Alexander Tolstoy hasn't got time to run for president. He's too busy not being arrested running around finding FOSS such as: *Darktable*, *Ddgr*, *Midnight Commander*, *Flameshot*, *Liquidshell*, *Posterizer*, *Android File Transfer*, *Catimg*, *Desktopfolder*, *Bernuse* and *HexGL*.

Next month..... 98

Protect thy privacy! Beat GCHQ and the NSA in one fell swoop and get yourself added to a watch list.



THIS ISSUE: Spectre & Meltdown » Netcloud Talk » Barcelona goes FOSS » RISC-V

PROCESSORS

Spectre & Meltdown flaws hit CPUs hard

Intel, AMD and ARM processors are all affected in a various complex ways, causing panic, confusion and much hand-wringing by those in the industry.

By far the biggest tech story recently (sorry useless robots of CES 2018) has been the discovery of serious design-level chip flaws that can potentially be found in the majority of processors in use today. The flaws were first found by Google Project Zero researcher John Horn, and Werner Haas and Thomas Prescer from Cyberus Technology in Dresden, Germany, and Daniel Gruss, Moritz Lipp, Stefan Mangard and Michael Schwarz from Graz University of Technology based in Styria, Austria.

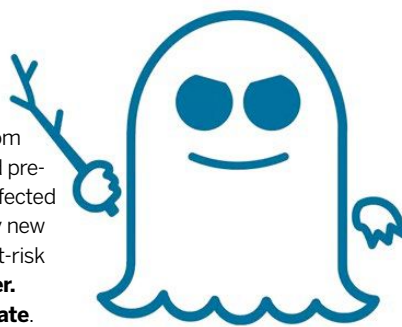
According to a timeline posted on The Verge (<http://bit.ly/verge-spectre>), Gruss, Lipp, Schwarz and Mangard discovered the fault late 2017, and on 3 December 2017 they had created a workable exploit for what would be called Meltdown, and contacted Intel. Apparently, Intel already knew about the issue, but asked the team to keep quiet. It wasn't until the beginning of January 2018 that both the Spectre and Meltdown flaws were made public.

Meltdown (CVE-2017-5754) is a flaw that can be exploited to read the contents of private kernel memory by an unprivileged user, essentially enabling a program to access the memory of other programs and the operating system – something it wouldn't usually have access to. Michael Schwarz tweeted a video of what an exploit for Meltdown could achieve (<https://twitter.com/misc0110/status/948706387491786752>).

The scale of the problem soon became apparent when it was revealed that all Intel CPUs with out-of-order execution since 1995 were potentially affected, apart from Intel Itanium microprocessors and pre-2013 Atoms. No AMD CPUs are affected by Meltdown and only certain very new ARM processors are. For a list of at-risk ARM chips visit <https://developer.arm.com/support/security-update>.

According to Intel, Meltdown can be mitigated via operating system updates, and patches have already been released for Windows and Linux. Apple was hesitant to reveal that its devices were at risk (its own ARM processor are affected), but it released MacOS 10.13.2 and iOS 11.2, which contain mitigations. For more details on Meltdown you can read the whitepaper at <https://meltdownattack.com/meltdown.pdf>.

Meanwhile Spectre (CVE-2017-5753, CVE-2017-5715) can be exploited to gain



» Spectre and Meltdown flaws have been found in almost all processors manufactured during the past 20 years.

released that contain mitigations (because any attack using Spectre could likely use JavaScript).

More information can be found at <https://spectreattack.com/spectre.pdf>, but because Spectre and Meltdown are so widespread it's likely your PC is vulnerable, so make sure your devices

“Spectre and Meltdown are widespread so it's likely your PC is vulnerable. Make sure your devices are patched and up to date”

information from other running processes. This flaw is harder to exploit than Meltdown, but it's also harder to mitigate and it affects AMD processors as well as CPUs by Intel and ARM. However, software patches for browsers and operating systems have been

are patched and up to date, and that any patches you do install are from trusted sources. Malwarebytes discovered a fake Meltdown and Spectre patch that deposits 'smoke loader' malware on the victim's Windows machine – read more at <http://bit.ly/smoke-loaders>.

COMMUNITY

Spectre & Meltdown: Linux devs respond

Linux machines are susceptible to the flaws in their processors, and distro makers were quick to issue patches.

The scale of the Spectre and Meltdown flaws are worrying, but the response by software developers has at least been reassuring.

Systems running Linux are vulnerable to the flaws, but the community has been working hard to mitigate the issues, with numerous kernel updates being released. For people running Linux distributions using the standard kernel on x86 hardware, you should make sure you have an updated kernel. If you're comfortable doing so, getting a release candidate (RC) update from the main kernel tree will enable you to have the most

up-to-date patches (at the time of writing 4.15-rc9 is the most current release). If you're sticking to stable kernels, make sure you have at least 4.14.15. If you're using LTS kernels, 4.4.113+ and 4.9.78+ are the ones to make sure you have installed.

For ARM64 hardware, 4.16-rc1 is worth installing, and if you're running Android, check out the common Android Kernel tree at <https://android.googlesource.com/kernel/common>. These patches include mitigations for Meltdown.

Spectre is a bit trickier, but in the middle of January Kernel 4.15-rc8 was released, which included the Retpoline coding technique created by Google to mitigate against the flaw. According to Google on its Security blog (https://security.googleblog.com/2018/01/more-details-about-mitigations-for-cpu_4.html), Retpoline is a binary modification technique that protects against "branch target injection" attacks. More can be read about Retpoline at <https://support.google.com/faqs/answer/7625886>.

Various distros have commented on the Spectre and Meltdown, as well as issuing their own mitigations, including Mint (<http://bit.ly/mint-fix>), Fedora (<http://bit.ly/fedora-fix>) and Ubuntu (<http://bit.ly/ubuntu-fix>).



➤ Android devices, such as the Pixel 2 XL, are susceptible to the flaws.

HARDWARE

Hardware makers react

Intel, AMD, Google and more attempt to repair the damage.

Despite Meltdown also affecting limited ARM chips, and Spectre affecting AMD and ARM as well, it was Intel that received the brunt of the bad publicity.

Intel initially claimed that any patches for the flaws wouldn't introduce significant slowdowns on hardware, and by the middle of January it had released firmware updates for 90 per cent of processors made in the past five years. It also vowed to work with software developers and other hardware manufacturers to eliminate the flaws. However, this did not stop three class-action lawsuits being filed in California, Indiana and Oregon against the company (<http://bit.ly/intel-lawsuits>), with more expected.

Meanwhile, in a blog post (<http://bit.ly/microsoft-reaction>) Terry Myerson, Executive Vice

President, Windows and Devices Group at Microsoft explained that patches for Windows machines for Meltdown and Spectre would have variable effects on the performance of those PCs. Some patches from Intel were also found to cause system instability in some cases.

It wasn't plain sailing on AMD's side, either, with the company slow to admit that its chips were susceptible to Spectre, with a class action lawsuit accusing the chip maker of keeping quiet about the problems. It also emerged that a Windows update meant to mitigate the issue was causing PCs running AMD chips to fail to boot (<http://bit.ly/amd-boot-failure>). If you're concerned about the hardware you're running, check out the complete list of CPUs affected at www.techrap.com/guides/complete-meltdown-spectre-cpu-list.

Comment

Linux isn't immune



Gustavo Padovan

“ The recent disclosure of Meltdown and Spectre hardware vulnerabilities were unprecedented in the history of computing. They affect a substantial portion of chips powering most of the infrastructure used today.

While software vulnerabilities can be repaired with an update, it's a different story when it comes to hardware, and the Linux Kernel community had a hard time dealing with the situation.

The mitigation for Meltdown came in the form of a fundamental change of the kernel memory management through the kernel page-table isolation (KPTI) patch set merged in 4.15-rc6, which isolates the kernel page table from the userspace page table.

Spectre, on the other hand, is much harder to fix, and while initial mitigation exists, more efficient solutions are yet to be developed. As its name says, Spectre may still haunt us for quite some time.

These issues may be just the first of their kind but they're already causing all of us to be exposed. Too many service providers and product companies have failed and will continue to fail at patching their kernels. Shifting all industries and sectors toward following the mainline Linux kernel closely is more crucial than ever.”

➤ Gustavo is the principal software engineer working at Collabora Ltd.

Comment

Back your bits up



Keith Edmunds

What data is critical to your business? Customer lists, employee details, next year's budget... they are all important, but they're not critical. If you lost them, there would be embarrassment, time wasted and loss of revenue for sure, but the business would survive.

The critical data is the data that, if it were lost, spells the end of the business. It's often the business's intellectual property (IP). It represents the value of the business to the shareholders, the investors, and it's what differentiates this business from the others.

For a web-hosting business, it's their clients' websites. For many research and high-tech businesses, it's a git (or similar) repository. For my business, Tiger Computing, it's a git repo and a wiki. What's yours?

That's the data that must be backed up. It needs to be backed up off-site, preferably to multiple, independent locations. It needs to be backed up automatically, every day or every hour, whatever is appropriate. And the restore from backup needs to be documented and tested regularly.

If you have to, you can ask your employees for their home addresses again. You can rebuild your prospect list. But if you have nothing to sell, then you're sunk.



Keith is the managing director at Tiger Computing Ltd (www.tiger-computing.co.uk).

SOFTWARE

Nextcloud Talk now open for business

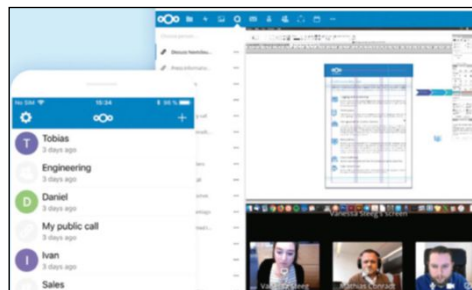
The open-source software provides end-to-end encryption on video calls, and negates the need for local installation.

After one and a half years of planning and development, Nextcloud Talk (<https://nextcloud.com/talk/>) has finally been revealed. This open source video meeting software is a secure software solution that's hosted on-premises, features end-to-end encryption and offers audio, video and text chat.

As a secure and open source alternative to Skype, it already appears to have some excellent features, including support to run inside a web browser, along with Android and iOS apps, with users not needing to sign up or install any software to join a call.

There are versions for home users and for businesses, with Frank Karlitschek, managing director at Nextcloud, stating that. "Business users have optional access to the Speed High Performance Back-end offering enterprise-class scalability, reliability, and features through a

Nextcloud subscription". As Karlitschek points out in a blog post announcing the release of Nextcloud Talk (<http://karlitschek.de/2018/01/nextcloud-talk-is-here>), the fact that it's self-hosted, and that all calls are handled by a user's Nextcloud server, makes the service unique.



Nextcloud Talk runs on PCs and mobile devices, and is an open source alternative to Skype.

OPEN SOURCE

Capital idea!

Barcelona city council wants to use open source software.

Barcelona's city council has announced that it aims to replace all software used by its employees with open source alternatives by spring 2019. At first it will keep Windows as the only proprietary software used, while it migrates users to open source software, but that too will be replaced with Linux. According to Spanish newspaper *El Pais* (<https://elpais.com>), 70 per cent of the city's software budget will be invested in open source software. For more information visit <https://publiccode.eu>.



Barcelona has become the first municipality to join the Public Money, Public Code campaign.

HARDWARE

Triple chip news

Esperanto Technologies plans for top-end RISC-V CPUs.

At the seventh RISC-V Workshop, Esperanto Technologies, which up until then had not been a well-known name, despite being backed by some big players, such as Western Digital, revealed it was planning to pave the way for RISC-V to enter the high-performance computing market with at least three RISC-V IP cores.

These are the ET-Maxion, a high-performance core set to rival the best ARM IP cores on the market today. The flagship core will aim to have the highest single-thread performance of ARM IP cores and is a 64-bit RV64GC processor.

Next is the ET-Minion, which is an energy-efficient core for high-teraflop computing, specialising in high-floating point throughput, and will hopefully reduce the energy cost of running the 64-bit RISC-V processor.

Finally, the ET-Graphics is a RISC V-based graphics processor capable of distributing workloads over a large number of cores. These cores have been made using the 7nm process by TSMC (Taiwan Semiconductor Manufacturing Company, Limited). To find out more, visit the website at www.esperanto.ai.



Distro watch

What's behind the free software sofa?

KAOS 2018.01

It's a new year, new ISO for the independent rolling distro with KDE Plasma. This latest version includes the Linux 4.14.14 kernel, which includes Retpoline and other mitigations against Meltdown and Spectre (see this month's main story for more details).

The very latest packages for the Plasma desktop are included, such as Frameworks 5.42.0, Plasma 5.11.5 and KDE Applications 17.12.1, which are all built on Qt 5.10.0. For more information, and to download this distro, visit <http://kaosx.us/news/2018/kaos01>.



› Update to 2018.01 if you're worried about Meltdown and Spectre.

NUTYX 10.0

Built from Linux From Scratch and Beyond Linux From Scratch, this French distribution (ne parle pas Français? It also supports multiple languages), comes in two versions: one a simple base image without a desktop, and one with an Xorg graphical environment.

The latest version, which was launched just after the 10th anniversary of the distro, comes with the 4.14.13 LTS kernel and a graphical front-end for package management called Flicards. To find out more details, visit its website at: www.nuttyx.org/en/news.



› A majestic logo for a majestic distro.

SOLYDXK 201801

This Debian-based distro with the KDE Plasma desktop has received a new snapshot that includes fixes for the Meltdown vulnerability, as well as Device Driver Manager (DDM), which has now been integrated alongside Debian Plymouth Manager. You can also now safely remove old kernel packages and add new partitions to **Fstab**. There's also a system configuration tool called SolydXK System Settings. The release announcement (<https://solydxx.com/new-solydxx-isos-released/>) contains more details.



› Another distro has been updated to mitigate Meltdown

PORTEUS KIOSK 4.6.0

This new update of the Gentoo-based distro for web-only kiosks is now available to download and comes with some major software upgrades. These include the Linux kernel 4.14.13, Mozilla Firefox 52.5.3 ESR and Google Chrome 63.0.3239.132. As the release announcement states (which can be read at <http://porteus-kiosk.org/news.html#180115>), "This release fixes the Meltdown attack and partially mitigates the Spectre vulnerability through updated CPU microcode and on the application level."

More patches are also expected to be merged as new information about Spectre and Meltdown emerge.



› If you're running Porteus on a kiosk, make sure you have the latest update install to protect against Meltdown and Spectre.

Comment

Embrace Azure



Clyde Seepersad

Over 40 per cent of the virtual machines running in Microsoft Azure are now utilising the Linux operating system. Couple this with the finding from the 2017 Open Source Jobs Report from The Linux Foundation and Dice that cloud computing is by far the most in demand skill among hiring managers, and the fact that Azure is one of the most popular public clouds, and it becomes clear that more professionals with both in-depth Linux and Azure knowledge are needed.

This is why The Linux Foundation has launched a new training course, Administering Linux on Azure, to make learning these skills more accessible. Whether someone is a Linux professional who wants to learn more about working on Azure, or an Azure professional that needs to understand how to work with Linux in Azure, this course will provide the requisite knowledge.

The course, available now, is taught by Sander van Vugt, a Linux professional living in the Netherlands and working for customers around the globe. It is offered online, and completely self-paced, so students can focus most on the knowledge areas most important to them. Those interested can learn more at <http://bit.ly/linux-on-azure>.

› Clyde is general manager, training and certification, at The Linux Foundation.

Mailserver

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

» No to Netflix

I spent this past month watching Amazon Prime Video using a Linux Mint 18.1 Live CD (so no HDD exposed to the Internet). I only had to install *npapi* to get around the DRM requirement. This setup worked for three of the four weeks during the one-month free trial, but then it stopped working for some unknown reason.

Perhaps it's time for a Linux version of a live CD that's specifically configured to work for Netflix or Amazon Prime Video without any configuration necessary. It would be very helpful to us users. Is there an article here for discussion?

Stephen Brent, via email,

Neil says: When you say *Npapi*, I presume you mean the Pipelight solution? That's been abandoned for quite a while now. Netflix and

Amazon require DRM elements to successfully play the contents from those sites. *Firefox* and *Chrome* support these out of the box, for better or for worse. I fired up our recent Ubuntu 17.10 Live Disc, logged into Netflix and enabled the DRM elements when requested and all seemed to work!

» Cover love

Loved the cover of the August issue (226) and the nod to George Lucas' first film. Nothing like comparing using Windows to the dystopian society depicted in that sci-fi classic.

Les B. Labbauf, via email,

Jonni says: Thanks Les, I think the artist enjoyed making it, too. We even seem to have been noticed on Wikipedia. I think that reflects more on the quality of Wikipedia than anything else. It also gave Neil an excuse to bore



» Why would working at Linux Format Towers make us dream of a dystopian future?

the *Linux Format* team with stories of the sound engineering used in the film.

» Keep escaping

I'm having a bit of a different issue with configuring my laptop for dual boot. My laptop is an MSI Gaming machine that came with a Plextor M6M (mSATA) 128GB SSD. I've added two more and have configured them as RAID5, handled by the BIOS UEFI. Windows 10 is installed on half of the drive and the other half is available.

When I boot with the USB Flash drive, I can start Ubuntu live with no issues. When I start *GParted*, I get a message that says the sectors are 2,048 in size and Linux only likes 512.

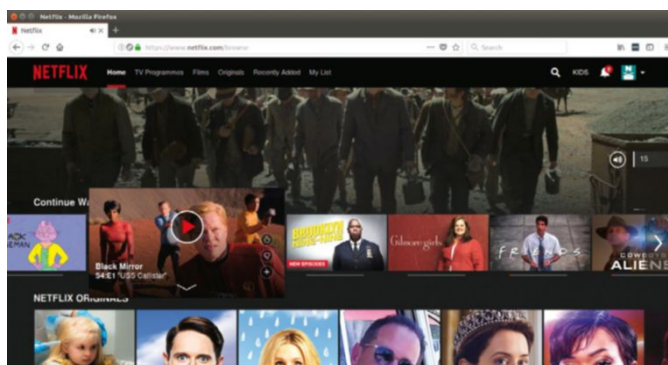
I tell it to ignore and I see all of my drives and the RAID. *Gparted* shows the non-allocated space and I'm able to access it.

I select it and issue the instructions as outlined in the article. To be sure that the settings are correct, I restart the machine. When I go back into *Gparted* the free space is showing as non-allocated. I've done dual boots with a single drive but not with a RAID configuration. Any suggestions?

John Martin, Terrell, Texas, US

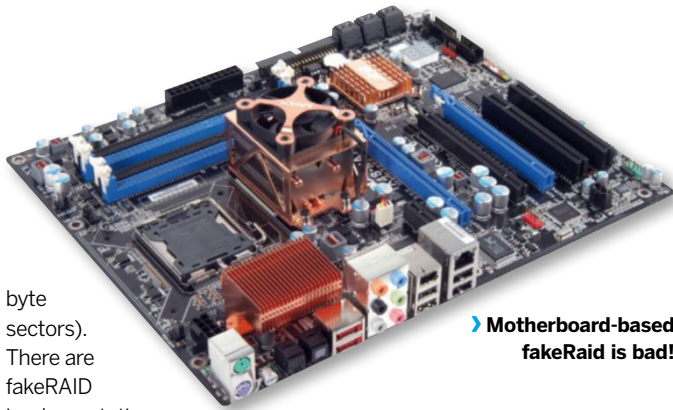
Jonni says: Unfortunately, the mechanisms used by motherboard firmware to do RAID (collectively known as 'fakeRAID') are generally proprietary systems, which is why *gparted* gets confused and can't do anything. It seems like you want to have both Ubuntu and Windows on the same array, but this will be complicated as Windows doesn't understand (or refuses to understand) Linux's softRAID.

FakeRAID isn't a good idea since, if your motherboard breaks then you may need to find the same model to gain access to your data again. I suspect the reason things disappear after a reboot is that your motherboard's RAID doesn't like what *gparted* is trying to do. Likewise, *gparted* doesn't understand what your motherboard is trying to do (it probably doesn't really have 2,048



» One advantage of Firefox supporting DRM out of the box is you can access your favourite streaming service from a Live Disc.





byte sectors). There are fakeRAID implementations that work in Linux, through *dmraid*, but I've never played with them and there's some conflicting information online. Hope you find a solution, and duly escape Windows.

➤ **Motherboard-based fakeRAID is bad!**

good to achieve in the long run I think. I suspect your answer will be, "You must be joking!"

Francis, West Cork, Ireland

Neil says: Yikes, that old Q6600 in itself will consume quite the measure of power – its TDP is 105W. A modern desktop processor will likely half the power (taking into account lower-power DDR4 memory and motherboard chipset) and still provide over double the processing power.

Combine that with (potentially) integrated graphics that's able to run three displays and with the right motherboard built-in dual NICs. The new AMD Ryzen 5 1600 is likely a good all-round choice to build from scratch, but you'll

need a graphics card on top. On the Intel side for similar performance you'd want a Core i7 7700K. It's £100 more just for the chip, although it does have integrated graphics. So it does sound like you want to build from scratch then?

I'm not going to be able to specify parts precisely, but I'm happy to point out a few I know of.

Yes, the PSU is important, as a poor 600W PSU will waste 30 per cent of the power, while a good one will waste just 10 per cent or less. Look for the 80 Plus Bronze measure, while a Gold will be worth it in the longer run.

The other option is to consider how you're handling your data? Is the download part the bit that takes the most time? Could that be handled overnight by a Raspberry Pi, laptop or miniPC box? Do let us know how you got on!

➤ LXF USA!

I live in Boston, and I hate paying more for *Linux Format*. I would like to pay less for an American distributed/published *Linux Format* magazine!

Markus McLaughlin, Boston, US

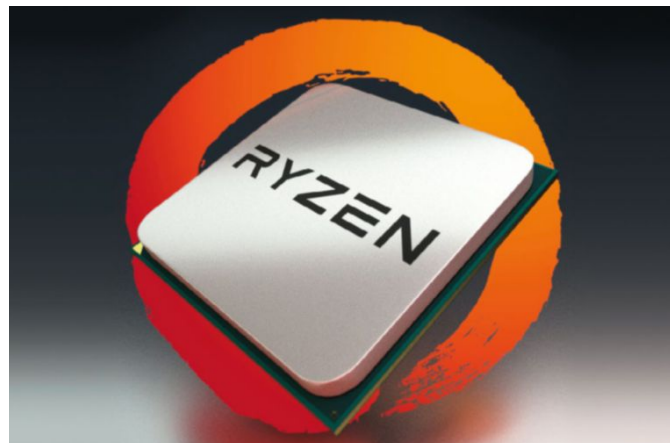
Neil says: While we'd love to launch a dedicated North American title, it's doubtful that'll ever happen. Print publishing is a declining industry and while niche titles such as *Linux Format* have been weathering the storm and maintaining sales, it'd be foolish to jeopardise the current stable business model with such a large shift. *Linux Journal* just dodged closing as it secured additional funding. That eye-watering cover price reflects the real costs of printing dead-tree tomes, shipping them around a continental-sized landmass, paying humans to write 60,000 words, produce a shiny DVD and put it all together with little to no advertising revenue.

➤ No Neon?!

Comparing KDE Desktops without including KDE Neon, the current champion of all things KDE, is slightly absurd. There are very few distros that can provide the latest up-to-date KDE on a stable LTS platform.

OpenSUSE can to an extent by adding additional repos, but that's about it. KDE Neon is designed to be slim with just enough to get you going, so users can install only the applications they actually want. I've been using it since KDE Neon 5.8 and it's pretty stable. I've only run into a few issues, like *Calibre* failing to install and needing upstream stable installation directly from Calibre.

After using KDE for years with Fedora and openSUSE KDE, Neon is clearly the best KDE experience I've had so far. The lack of YaST or other third-party system tools does demonstrate the need for more community-driven system configuration tools and KDE has delivered one

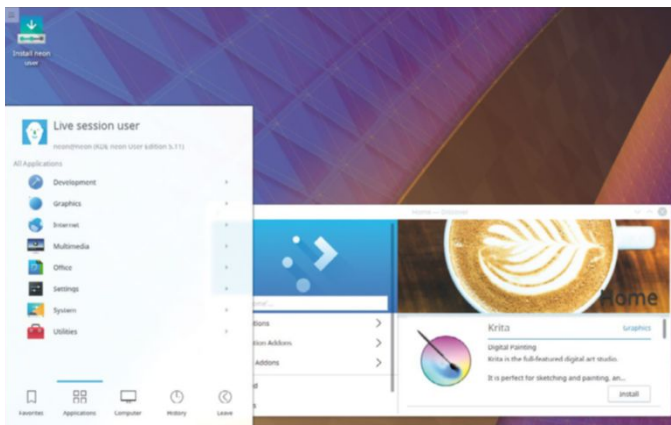


➤ **The AMD Ryzen CPU offers competitive multi-threaded performance.**



Write to us

Do you have a burning Linux-related issue you want to discuss? Want to ask for upgrade advice, let us know what we're not covering or just want to tell us what a wonderful bunch we are? Write to us at *Linux Format*, Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or lx.f.letters@futurenet.com.



» The KDE Neon distro is damn funky!

» for Systemd with hopefully more to come in the future.

Timothy Butterworth, via email

Jonni says: Thanks for your message. I, too, am a fan of KDE Neon. It was a *Roundup* of KDE distros though, and Neon's own FAQ points out that it is 'not quite' a distro, since no software is bundled beyond KDE bits.

While this won't be a problem for a lot of people (and in fact people like you and I quite like it because we can tailor it to our own purposes), it means it can't cater to those that want everything ready to go out of the

box, and makes comparisons to other, actual distros hard. It probably did deserve more of a mention though.

But hopefully the KDE feature that ran in **LXF229**, gave Neon the credit that it is due.

» Android is Linux

I was very surprised and disappointed to see *Linux Format* fostering the myth that Android is an alternative to Linux, rather than a type of Linux. This appeared in two stories in **LXF226** (August 2017). First, in the Newsdesk section,

repeatedly in the main story about smartphones; and also in the last Newsbytes item, which refers to "getting Linux to work on Android."

I'm used to seeing that mistake in the mainstream media, but you folks?

Gregory Miller, via email

Neil says: Thanks for your letter and we do like having our mistakes pointed out as it helps us improve the magazine for next time. However, with this Android point I'm a little confused as I'm not sure we're doing what you're accusing us of.

The main news story in **LXF226** is about how the failure of Ubuntu Touch has enabled UBTouch to take its place, while also looking at a couple of other FOSS projects aimed at mobile devices that use the Android ecosystem – typically utilising the often proprietary drivers in the Android HAL – to enable bringing fuller Linux distros to those mobile devices like Debian.

The smaller story is about a software tool that enables you to run Android apps on desktop Linux distros. I think it's largely as we're using Linux here as a lazy

shorthand for GNU/Linux, in other words a full desktop distro, rather than Android/Linux which are bespoke builds for single hardware devices with a limited update life and contain proprietary parts.

Android is a fine Linux-based mobile OS. It's not a FOSS GNU/Linux distro, which is what the news article was about. **LXF**



» Most Linux Format readers aren't about to use Android as their day-to-day desktop driver.



Letter of the month

Pendulous!

When it all started, a computer cost a couple of million 1960s US dollars, and a company or institution generally could afford only one. Everybody had to take turns using it, often in the wee hours of the morning. Then came timesharing, dumb remote dialup terminals and operating systems that enabled multiple users to access the computer simultaneously.

Then came the minicomputer: less powerful but cheaper by a bigger factor, and a department could afford one, so it was back to taking turns, but at least by a smaller group. The minis became timeshareable by a few users, but soon after came the professional desktop workstation and the less-powerful but much cheaper desktop personal computer. The paradigm of the day was an unshared computer for everybody and no waiting.

But that made it hard to share data, and so ensued an ever-changing series of networking technologies, starting with dial-up BBSs and going from there, to the world wide web, initially for somewhat limited purposes, then generalising to cloud computing for everything.

We were centralised again, and sharing was easy. It also meant not nearly so much local power was needed, so cheaper, less-powerful thin clients and netbooks appeared at the users' end.

Now, the latest paradigm shift is edge computing, where we move away from centralised, shared computational, storage and networking hardware resources, and place more power at the edge. Less waiting and maybe cheaper hardware? Does anybody notice the pattern? It's only because I'm old enough to have seen it all? Will it stabilise some day?

Rodney Bates, Strong City, KS, USA

Neil says: Thanks for your insightful letter. I think you're spot on here, what I would add is edge computing is taking advantage of the new generation of ultra-low cost ARM-based SoCs that are enabling networked data collection in areas that wasn't possible before. But it's all part of the wonderful cycle of invention, innovation and implementation. It'll be interesting to see what effect of the next-generation of ultra-low power use ARM servers have on the market.

» Canonical is enabling projects to make better use of edge computing innovations.





United Linux!

The intrepid **Les Pounder** brings you the latest community and LUG news.

Find and join a LUG

» Alpinux, le LUG de Savoie

Meet on the first and third Thursday of the month at the Maison des Associations de Chambéry.

www.alpinux.org

» Build Brighton

Thursday evenings is open night.

www.buildbrighton.com

» Sandbox Digital

5 Brasenose Road, Liverpool. Open maker night is Tuesday 6-9pm, kids clubs are Monday (six to eight years), Wednesday (eight to 12).

www.sandboxdigital.co.uk

» Leeds Hackspace

Open night Tuesdays, 7pm-late, Open day second Saturday of the month, 11am-4pm

www.leedshackspace.org.uk

» Horsham Raspberry Jam

Park Side, Chart Way, Horsham, West Sussex.

www.facebook.com/hackhorsham

» rLab Reading Hackspace

Unit C1, Weldale St, Reading. Open sessions Wednesday from 7pm

www.rlab.org.uk

» Huddersfield Raspberry Jam

Meet every month at Huddersfield Library, typically the fourth Saturday of each month.

www.huddersfieldraspberrypjam.co.uk

» Medway Makers

12 Dunlin Drive, St Mary's Island, Chatham ME2 3JE

www.medwaymakers.com

» Cornwall Tech Jam

Second Saturday of the month alternating between Bodmin and Camborne

www.cornwalltechjam.uk

Pay it back with interest

How will you contribute to your community?

As long-time *Linux Format* readers will know, I was once the Chief for Oggcamp and every year my team of volunteers would ensure that those coming to Oggcamp would have a fantastic experience that weekend.

So we just turned up on the day and looked cool, right? Of course not – it's never that easy and there were many weeks of conference calls, venue visits, to-do lists and so on. Then we looked at crew skills: were there any first aiders, who could drive, or spoke French or German. The organisation was endless, whereas the available time was not.

So why did we do it? Well, I wanted to give something back to the community that had nurtured me for so many years, and that sentiment was shared with those who volunteered their free time, and spent their own money to come to the event and help out. Sure, they got a T-shirt and a mug for their help, but the richest reward was helping their community to be the best that they can be.

For many people, the idea of contributing to a community – be it Raspberry Pi, Python, BASH or Perl – is via code. But for those who can't write a line of code, there are many more ways to help grow what you love. Organising events such as bug-squashing parties or talks at your local user group meeting is one great way to start giving back with coding! If you have a journalistic flair then helping to document an open source project is a worthwhile exercise and it could even lead to a career!

The moral of this story? Give back to your community. It's the greatest expression of gratitude we can make to those that give up their time. **LXF**



» The Oggcamp crew is a wondrous group who give up their free time to help the community!

Community events news



Oggcamp 2018

This year's Oggcamp's host city is... Sheffield! It's taking place at Sheffield Hallam University on 18 and 19 August. This venue is also the home of Steelcon (a hacker conference). Over the Oggcamp

weekend you can expect talks covering the wide interests of the community and stalls where you can learn first hand about new open source tech and products. There'll be more information on the website soon!

www.oggcamp.org

Red Hat Summit

The Moscone Center and Marriott Marquis in San Francisco, California plays host to Red Hat Summit 2018, on 7–10 May. This event is a mixture

of hands-on workshops, "power training" with experts, breakout sessions and talks for engineers of all levels. This event isn't cheap – early bird tickets are \$1,300 and go up to \$1,600 and \$1,800 if booked nearer the time – but you'll receive four days of intensive training from one of the most important organisations in the Linux community. Tickets, agendas and more details can be found on the website.

www.redhat.com/en/summit/2018

Electromagnetic Field 2018

Music lovers have Glastonbury, but makers have Electromagnetic Field! This camping festival takes place from 31 August to 2 September in Eastnor, Herefordshire and offers talks, workshops and hands-on demonstrations. If you'd like to learn textiles, ironmongery, software-defined radio hacks and more, then this is the event for you. Tickets and more information via the website. www.emfcamp.org



Ask Dr Bidwell

Answers

Got a question about open source? Whatever your level, email it to lxf.answers@futurenet.com for a solution.

1 Qubes questions


Q I've just installed Qubes 3.2 and have updated it, yet I can't seem to figure out how to play any video (from news stations, movies, DVDs). I realise special hardware is called for, but I have it all: my CPU is an Intel i7-2760QM running in an HP 8460p laptop, with SSD and 16GB RAM. I have 14 years of experience with Linux.


I use Linux Mint 18.3, and I have all the requirements for Qubes, but can't seem to get video to work. I can't download the Flash plugin and I don't know much about HTML5. I don't know how to install/extract Tar files. If you can help, thank you. If you can't thanks anyway. I still plan to continue using Linux Mint – it does everything I ask of it. But Qubes does have many advantages.

Stephen Wood, Via email


A Qubes (as you're no doubt aware) is focused on privacy and security. Applications are all run in separate Xen domains that can't access each other and have limited access to the host's hardware. So your multimedia applications can't see your video card, only a virtual video device that painfully renders everything in software before passing the results to the host, which then throws it all at the real video card.


Still, I'd expect playing local files (low-res ones at least) with something like VLC to work through software rendering, especially given the potency of your hardware. In fact, I'd expect streaming video from the web to work too, albeit with a bit of jitter and stutter. I haven't played with Qubes for a while, and I don't have



"If you're serious about security, Qubes OS is the best OS available today. It's what I use, and free." 

— Edward Snowden, *whistleblower and privacy advocate*



"When I use Qubes I feel like a god. Software thinks that it's in control, that it can do what it wants? It can't. I'm in control." 

— Micah Lee, *Founder, Coderabbit Foundation*

➤ If it's good enough for Ed and Micah, then it's probably good enough for you. Unless you want to watch high-definition cat videos all day long.

an installation to hand (it doesn't work well in a VM for obvious reasons), so this is just speculation. Which OS template are you using with Qubes? It could be a distro specific issue. Are there any relevant error messages? Google is your friend here.

There's really no good reason to install the Flash plugin nowadays (especially if your motivation for using Qubes is security). Almost all sites now offer some kind of HTML5 video, and even DRM-protected content is supported out of the box in Firefox. It's possible that you're missing some library (*libav* or *ffmpeg*) to make this work, but then we'd still need an explanation for DVDs and local files refusing to play on your laptop.

I'm reliably informed that we do have a privacy feature (*you're supposed to be writing it now – Ed*) coming up next issue, so hopefully I'll get a chance to play with Qubes. I'll be sure to include any tips on video playback if I come across something.

You can extract the contents of a tar file with `tar xvf file.tar`. Most tars are gzipped (or xz-ed or bz2-ed) to make them smaller, *tar* (tape archive) doesn't do compression, so pass the archive to *gunzip* first do `tar xvzf file.tar.gz`. More details available via `man tar`. As an aside I'm slightly confused that you could go 14 years without having cause to do this. Obvious warning about installing unverified binaries instead of using a package manager.



Esfuerzos para esquivar EFI

Q I have a Mac Mini from 2006 which no longer receives updates from Apple. I hope to make this machine great again by installing Linux, but I've not made much progress.

I've tried to boot several of your DVDs on this, but the machine just gets stuck, with a message saying "Select CD-ROM Boot Type". I understand that there are some issues booting Linux due to these machines' 32-bit EFI, but I can't think of an easy solution. Any ideas?

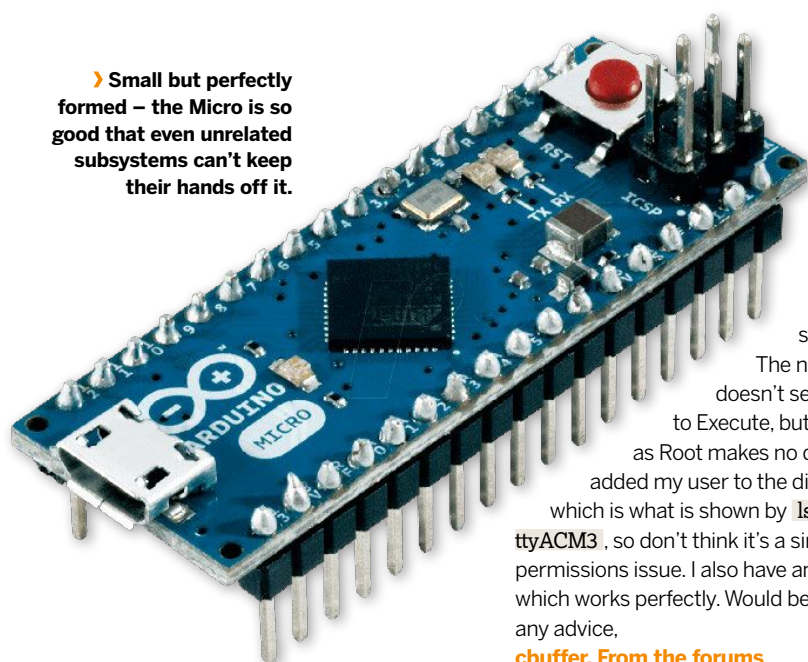
Pedro Perez, Puebla State, Mexico

A Is that you Effy? There were two Mac Minis that came out in 2006, and you're right, both have a 32-bit EFI. However, the newest firmware enables them to boot in BIOS compatibility mode, which is also needed for accelerated video in Linux, so updating firmware is your first task.

Sadly, even this won't make it possible for you to boot the multicatalog images that most 64-bit Linux distros (and our DVDs) use. The early 2006 Minis had 32-bit Intel Core Duo CPUs, and the late ones packed 64-bit Core 2 Duos. If yours is the early 2006 model, then

your only solution is to boot a 32-bit distro. For the later edition, you can re-engineer a standard 64-bit distro, ISO stripping out the UEFI booting support. Essentially you just mount the ISO, and remaster it with *xorriso* using just the boot code from the syslinux package (see for example the section on Removing UEFI boot support at https://wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface). You'll likely want to install the *rEFInd* boot manager if you're dual booting, and there may yet be other obstacles to surmount. Good luck!

Small but perfectly formed – the Micro is so good that even unrelated subsystems can't keep their hands off it.



gives the same error.

The native install doesn't set permissions to Execute, but changing that as Root makes no difference. I've added my user to the dialout group, which is what is shown by `ls -l /dev/ttyACM3`, so don't think it's a simple permissions issue. I also have an Arduino Uno, which works perfectly. Would be grateful for any advice,

cbuffer, From the forums

There are a few references to this problem online (see <http://starter-kit.nettigo.eu/2015/serial-port-busy-for-avrdude-on-ubuntu-with-arduino-leonardo-eth>, for example). Some people have reported success by repeatedly resetting the Arduino and reattempting the upload, but this seems like an unsatisfactory solution.

What's transpiring is a conflict with the modem-manager package for managing mobile broadband (2G/3G/4G) devices. The *modem-manager* (hereafter *mm*) daemon claims the device nodes, because it assumes anything connected over a serial link (the `/dev/tty*` nodes) is a modem. This prevents the *avrdude* application from connecting with them, causing the error you quote.

The solution in the example mentions adding a udev rule which sets the `ID_MM_DEVICE_IGNORE` variable to stop *mm* claiming

these nodes. Oddly enough though, Ubuntu 16.04 and derivatives (such as Mint 18.1) already include such a rule (in the file `/lib/udev/rules.d/77-mm-usb-device-blacklist.rules`). It uses Arduino's USB vendor id (2341) to identify the device and set the environment variable if it's detected, so in theory this problem shouldn't exist anymore. And yet it persists, so there must be some bug or other process that's letting *mm* dig its claws in (you can check *mm* is indeed the culprit by plugging in the Micro and running `lsdf /dev/ttyACM*`). If you don't use any mobile broadband devices, then the solution is simple: Just nix the obstreperous package with `sudo apt remove modem-manager`. You can always re-install it later if the need arises. If you do need *mm* though, then please let us and so many other perplexed Arduino users know if you find a less-brutal solution. **LXF**

2 Argh-duino

Hi Team *Linux Format*, just hoping there are some Arduino users here. I'm running Mint 18.1 Cinnamon on an AMD FX-4100. I recently bought an Arduino Micro to use with a Pi-Zero. I initially did `apt-get install arduino` but some of the files appeared to be corrupt. I've since downloaded them from the Arduino site twice and despite following its install and troubleshooting advice I get the following error message when starting the basic blink sketch:

```
avrdude: ser_open(): can't open device "/dev/ttyACM3": Device or resource busy
```

An error occurred while uploading the sketch

I've obtained the same result with a laptop using Mint and an old laptop running XP. Maplins accepted the Micro may be faulty and replaced it with another, which embarrassingly

Get help now!

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

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

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

```
lspci >> system.txt
```

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



Festive tales of resurrection

I've had four Ubuntu desktops, but Santa didn't bring me a Linux laptop for Christmas. I then remembered that about a year ago, an old friend gave me his beaten up (well, just superficial wounds, really) Dell Studio laptop running Windows 7. It took several minutes to boot up and was crammed with lots of annoying pop-ups. I used the DVD that came along with LXF231 to test 32-bit Ubuntu 17.10 on this Dell laptop and was amazed at the speed of operation.

I had just two problems, now happily overcome. First, I found that it's really difficult to remove a DVD from a laptop slide-in slot after you've formatted the HDD and deleted the old OS. Anyway I managed to install Ubuntu from the

imprisoned DVD and I'm impressed with the results. Oh, the second problem is that the laptop's Wi-Fi stopped working. I borrowed an external Wi-Fi dongle from a smart TV and plugged it into a USB port on the laptop. This enabled me to download a driver for the internal Broadcom Wi-Fi card and then (joy of joys) I saw the Wi-Fi LED on the keyboard light-up. I now have the Linux laptop that I craved, but Santa didn't bring me. I love the minimalist Ubuntu desktop and the thin toolbar along the bottom edge, and all the apps are conveniently arranged into categories. No more scrolling down the Ubuntu launcher for me!

So, let me send a big thank you to the good folks at *Linux Format* magazine (and

not forgetting my friend who donated his old damaged laptop). Between you all, you've made a 70-year old man very happy this Christmas.

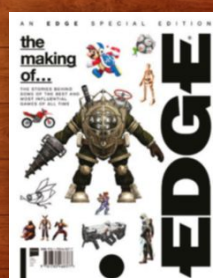
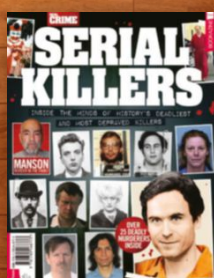
Pete, via email

Thanks Pete. Glad we could contribute some festive cheer. That Santa has much to answer for. In the old days optical drives used to all feature an emergency eject mechanism that you could prod with a handy paperclip. Slot-loading drives tend not to bother with these, which means you need to load an OS to eject the drive, leading, as you discovered, to some frustrating Catch 22 situations. As mainstream offerings move away from 32-bit support it'll be interesting to see how distros like Ubuntu and Bodhi evolve.



Discover another of our great bookazines

From science and history to technology and crafts, there are dozens of Future bookazines to suit all tastes



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



www.myfavouritemagazines.co.uk

Magazines, back issues & bookazines.

LINUX FORMAT Reviews

All the latest software and hardware reviewed and rated by our experts

AMD Ryzen 3 1200

Alan Dexter learns that the most budget-conscious spin of Ryzen highlights the multicore norm and performance delta of CPUs of old.

Specs

- » **Socket:** AM4
- » **Type:** 64-bit
- » **Process:** 14nm
- » **Cores:** 4
- » **Threads:** 4
- » **Clock:** 3.1GHz
- » **Turbo:** 3.4GHz
- » **Cache:** 8MB L3, 2MB L2, 384KB L1
- » **Mem:** DDR4, two channels, 64GB max
- » **TDP:** 65W
- » **PCIe:** 20 lanes
- » **Virtual:** AMD-V, VT-Vi, 2x AES

As the song goes, you can't always get what you want, and in lieu of being able to afford a top-flight monster, a budget chip like this will have to do. But Rolling Stones references aside, it's not the whole story when it comes to AMD's most affordable Ryzen chip. This isn't just a chip to turn to when forced to build on a shoestring. It's a chip that shows how far CPUs have advanced in a year, and it's a chip that punches far harder than its £100 price tag suggests.

That single ton of cash nets you a true quad-core processor. Yup, the sort of territory that was previously the preserve of a mainstream Core i5. The Ryzen's base clock nominally runs at 3.1GHz, but you'll rarely see it running the cores at that speed; instead it'll operate closer to 3.4GHz. There's no simultaneous threading included on

AMD's Ryzen 3s, presumably to help separate them from the Ryzen 5 chips, but it's still a speedy chip, capable of handling four threads at once.

We've compared it to the eighth-generation Core i5-8400, purely because that's the chip to beat in the desktop space at the moment. On the face of things, that's not a fair comparison. The new Core i5 is a six-core, six-thread chip, but it does cost double what AMD is asking for this, so bear that in mind when looking over the specifications any benchmark results.

Steady as she goes

Speaking of benchmarks, we're pleased to report that there are no nasty surprises with the Ryzen 3 1200. Given the core count and operating frequency, it performs as expected.

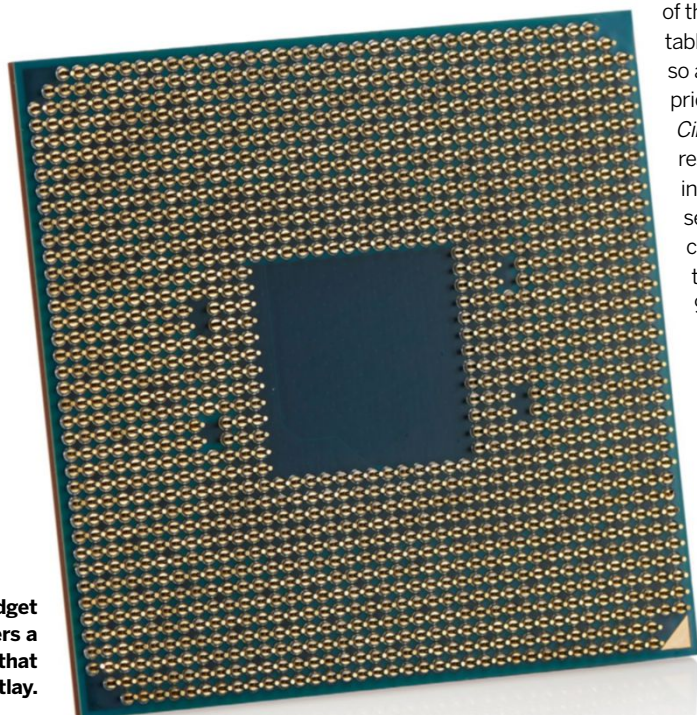
This means that it's bringing up the rear of the benchmark tables, but it does so at a great price. The *Cinebench R15* result of 478, for instance, may seem low compared to the Core i5's 949, but in the context of price, you're looking at a score of 4.78 per pounds, as opposed to 4.75 for Intel's. You

get more bang for your buck from AMD's chip.

It's worth noting that the latest BIOS updates mean that the memory issues we saw at the launch of Ryzen have evaporated, and that getting the system up and running was a breeze.

There's a Wraith Stealth cooler in the box alongside the CPU, and it's a fine cooler at stock performance. It's worthy of its "Stealth" moniker, too. When it came to overclocking, we reached for our AIO cooler of choice – the NZXT Kraken X62 – but you should be able to hit decent numbers with the Stealth as well. Overclocking the Ryzen 3 1200 is simple: set the target frequency in the BIOS (increase the voltage if needed), and you're done. In this case, we hit a stable 3.7GHz, which saw the *Cinebench R15* score jump to 567 points.

Overall, this is a lot of chip for not much money. If you need serious power, you should absolutely spend more, but if value is your main concern then there's certainly a lot to love here; the proviso being until the GPU-equipped APU Ryzen units appear. **LXF**



» AMD's budget chip offers a lot for not that much outlay.

LINUX FORMAT Verdict

AMD Ryzen 3 1200

Developer: AMD
Web: www.amd.com
Price: £100

Features	8/10
Performance	7/10
Ease of use	9/10
Value	9/10

» Until we see graphics-capable APU desktop versions of the Ryzen, this entry-level processor represent solid value for money and performance.

Rating 8/10

Intel Core i3-8350K

Driving the budget concept in a whole new direction **Alan Dexter** struggles to keep up, doing a three-point turn in Bath's traffic.

Specs

- » **Socket:** 1151
- » **Type:** 64-bit
- » **Process:** 14nm
- » **Cores:** Four
- » **Threads:** Four
- » **Clock:** 4GHz
- » **Cache:** 8MB
- » **Mem:** DDR4, two channels, 64GB maximum
- » **TDP:** 65W
- » **PCIe:** 16 lanes
- » **GPU:** Intel UHD Graphics 630
- » **GPU Clock:** 350MHz (1.15GHz max)
- » **Virtual:** VT-x, VT-d, EPT

There are two important factors when it comes to Intel's Core i3-8350K. First, it's unlocked; and second, it costs £160. It's good that you can overclock it, but that's unquestionably a lot of money for a chip that most of us would perceive as being a budget offering. For context here, this isn't the only eighth-generation Core i3 currently available, as the Core i3-8100 rolls in at a much more palatable £100. Intel clearly knows what sort of price it can expect to charge for budget hardware, even if it's decided to ignore that wisdom here.

Under Intel's new branding, a Core i3 is a quad-core processor that lacks Hyper-Threading (just like the Core i5), but also lacks a Turbo mode. To be fair, the base clock frequency of 4GHz is healthy enough, even if it won't be jumping up and down as more/fewer cores are used. As we've mentioned, though, this CPU also happens to be unlocked, so if you want to push the chip harder, you can. And you really can.

You get 8MB of cache to help keep things ticking along nicely, support for up to 64GB of DDR4 RAM, and integrated graphics in the form of Intel's UHD Graphics 630, which has a nominal base clock of 350MHz, capable of hitting a maximum speed of 1.15GHz. While this obviously can't compete with dedicated graphics silicon when it comes to driving the latest games, it does mean that you can build a machine without a discrete GPU, which can help keep the overall price and size of the machine down, versus a Ryzen system, say.

There is one small problem when it comes to that notion of budget, though, and it's that your options on the motherboard front are limited to a single chipset, and that's the enthusiast-class Z370. There is a surprisingly good spread of options here, but with even the cheapest starting out at £90, we're some way off the £55 starting point that formed the basis of many a budget Core i3 build using the B250 mobos from the previous generation. More budget-conscious chipsets should be on the way, but for now, the combo of this chip



» This chip is a strong performer, but the price point is disappointing.

alongside the cheapest Z370 board starts out at just under £250. Gulp.

Performance part

We may have reservations about how this chip is being pitched, but when it comes to performance, our opinion is much more straightforward: it's great. That high-base clock speed combined with four real cores makes for some great results. Indeed, in testing, this chip was just a shade off the performance offered by the last-generation Core i5-7600K. Comparing it to the closest priced chip from AMD, the Ryzen 5 1500X, which is a quad-core chip as well, albeit with SMT (Simultaneous Multi-Threading), so it can handle eight threads. It's a neck and neck fight, with the added threads of the Ryzen helping it to win in some areas, but the raw grunt from Intel's single-core performance trumping Team Red (that's AMD) elsewhere.

When it comes to overclocking, we managed to get our silicon running at 4.9GHz, with only a little extra voltage (1.4V), resulting in a *Cinebench* score of 784 (with a single-thread score of 205).

Impressive figures, even if you'll need to spend more on your cooler to hit that.

The problem for this unlocked Core i3 is Intel's own Core i5-8400, a chip that costs just £40 more, yet boasts 50 per cent more cores. The Core i5 also has more cache and a lower TDP, at 65W as opposed to 91W. Of course, you can't overclock that chip, which is a win for the Core i3-8350K, but we'd still prefer to have the extra cores of the Intel or extra threads of the Ryzen. **LXF**

LINUX FORMAT Verdict

Intel Core i3-8350K

Developer: Intel
Web: www.intel.com
Price: £160

Features	7/10
Performance	8/10
Ease of use	9/10
Value	6/10

» For performance this is a fantastic budget processor, but it's priced as a mid-range device, which it really isn't.

Rating 7/10

Freespire 3.0

The word revival is rarely used in conjunction with dead distros, but Freespire has done just that thanks to new management, discovers **Shashank Sharma**.

In brief...

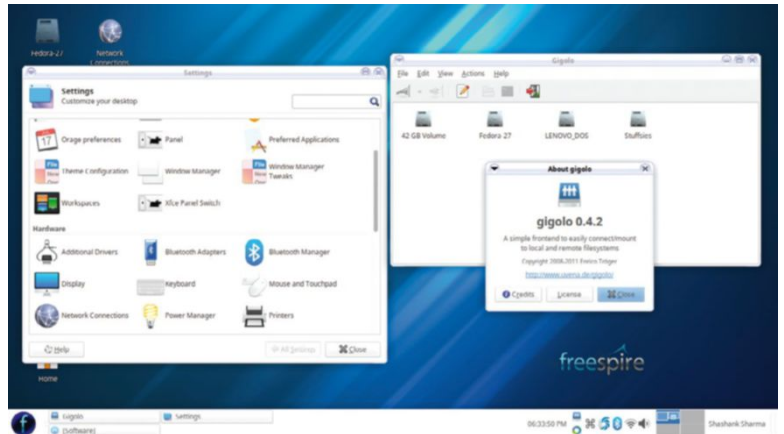
» Freespire was originally called Lindows and quickly became popular with users looking to move away from Windows, but not its appearance. Based on Ubuntu and powered by XFCE, the latest iteration is aimed at new users. Ubuntu, Linux Mint, Mageia and most popular distros are comparable alternatives.

Even with its first release, the distribution quickly managed to polarise the Linux community with its design, pitching ease of use as its greatest feature. No, we're not talking about Ubuntu, but the original Lindows OS, designed to emulate the look and feel of Windows. With its last release in 2007, the Linspire/Freespire distros died a quick death after being taken over by Xandros in 2008. A decade later, news broke of the release of Linspire 7.0 and Freespire 3.0, now owned and developed by PC/OpenSystems LLC, the group behind Black Lab Linux.

The original Lindows underwent major upheaval in the years since its launch, including a change of name and management. The project was also one of the first to offer a free desktop distro, named Freespire, along with a commercially licensed variant offering support under the brand name Linspire.

The subject of our review is Freespire, which is the same as the commercially offered Linspire, stripped of its proprietary bits and blobs and lacking some other popular software. The distro is available as a Live-installable ISO for 64-bit machines and features tools and software released under the GPL, GPLv3, BSD and a slew of other open source licenses.

At just 1.5GB, the distro is sparse. Its default software offerings feature none of the bulky apps that are the mainstay of most modern distros. For instance, Freespire doesn't ship with *Gimp* or



» Along with XFCE desktop environment, the distro features Geary email client and other lightweight tools to deliver a speedy performance.

even a complete office suite. Offering *DejaDup* backup tool and *MintNanny* domain blocker out of the box, along with a host of everyday internet and multimedia apps is a welcome decision, but some of its choice of default apps is puzzling. The distro ships with two orthodox file managers along with *Thunar*. There's also a tool to help you set up a PPP connection!

It's a bug's life

Despite being in development since June 2016, the distro ships with some minor bugs. Although it doesn't report any errors, the *MintNanny* domain blocker doesn't work as advertised. You can add and remove domains easily enough, but one can still easily access the websites. Also, you won't find the Settings Manager under the Settings sub-menu. Instead, you must click the button to the right of the search bar at the bottom of the launcher. We also found the Search bar to be fast and effective, which doesn't explain the inclusion of the *File Searcher* app, which is designed to help users find their favourite from among the installed apps.

While the latest release is based on Ubuntu 16.04 LTS and will be supported until 2021, the project doesn't provide any information about upcoming releases or features the project is working on, apart from a list of dates for future Linspire/Freespire releases on the Linspire website. For all intents and

purposes, Freespire is a completely new distro, with little to tie it to previous releases except the name. In such a scenario, it's imperative the project reassures users that it will be around for a while and not just disappear yet again.

Worse still, Freespire doesn't host any documentation on the website, apart from a two-page PDF installation guide. While this is to be expected since the project sells commercial support solutions, there also isn't any means to connect with the user community.

Although usable, its current release offers no reason for users to switch from their current Linux distro. Furthermore, the lack of information on future releases is another reason why we would advise users to adopt a wait-and-watch policy. **LXF**

Features at a glance

Esoteric tools

The distro features Worker file manager and Synergy, designed to share mouse/keyboard across machines.

Ease of use

The choice of Ubuntu makes Freespire easy to use and you also won't want for general support.

LINUX Verdict

Freespire 3.0

Developer: PC/OpenSystems LLC
Web: <http://bit.ly/freespire-3>
Licence: Various open source

Features	7/10
Performance	7/10
Ease of use	7/10
Documentation	1/10

» It's lightweight and fast, but that's relatively common. Lack of clarity over future releases will hurt its adoption.

Rating 6/10

Siduction Cinnamon

The premise of an entire distribution based on unstable software is seducing enough for **Shashank Sharma**. But is the project just a clever gimmick?

In brief...

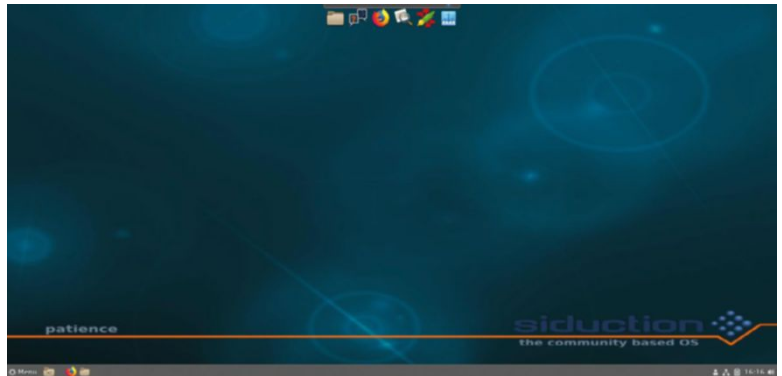
» Based on Sid, Debian's unstable branch, the distro presents itself as fully usable, if its default collection of apps is any indication. The caveat of running unstable software is repeated throughout the release notes, on the project's website, as well as during installation. Much like rolling release distros like Arch, you can use dist-upgrade to bring your system up to date.

You know you're at the heart of the Linux ecosystem, when even your unstable branch spawns a Linux distribution; based on Debian Unstable, Siduction is a Linux distro designed for users who are eager to run the latest software and also don't want to repeatedly go through an entire installation every six months.

The project ships as a sub-2GB Live installable ISO for 64-bit machines and features a customised Calamares installer. Unlike many Debian derivative distros, which are driven by a single developer or a small team, Siduction has a comparatively vast team of developers and maintainers.

Debian, which is well known for its Debian Free Software Guidelines (DFSG) – a set of rules that describe what software and tools can be included in the Debian release based on the underlying software license – has taken an exception to Siduction's decision to ship with various proprietary codecs and drivers out of the box. What's more, users can't choose to opt out of these proprietary offerings. You can find a list of such tools in the release notes, or alternatively run the `vrms` command from a terminal. The distro used to offer a custom script named `remove-nonfree` to rid your installation of all non-DFSG-conforming software, but has since discontinued it. You must run the `apt purge $(vrms -s)` command instead.

As with past releases, the distro ships with several variants featuring



» Despite the inherent risk of running unstable software, the convenience and ease of use of Siduction cannot be overstated.

different desktop environments such as XFCE, Cinnamon, Mate, KDE and Gnome. For users who aren't keen on a GUI, there's a noX version that ships without X. However, Gnome, Mate and LXDE will likely be dropped for future releases of Siduction, unless the project finds maintainers for these editions.

Calm and composed

Siduction does well to offer a slick experience to its users. This is evident from the custom Calamares installer, which now uses `Kpmcore` as its partitioning tool. The distro also features several custom tools, such as the aptly named `Activate SSH` and `Deactivate SSH` scripts. Then there's the `simple-paste` script, which the developers have dubbed as the Swiss Army Knife of pasting. Also featured are useful and uncommon tools such as `G Alternatives`, which can be used by administrators to define the tools that provide specific services to users.

For a project that's made entirely of software taken from Debian's unstable branch, we didn't find Siduction's latest release to be buggy or unusable. The lone exception was not being able to use the mouse scroll to read through the Release Notes during the installation process. The mouse works flawlessly throughout the rest of the installation steps.

The boot screen when running the live disc is slightly confusing. You must select the option to boot 'From CD/DVD/ISO', or the option to boot 'From

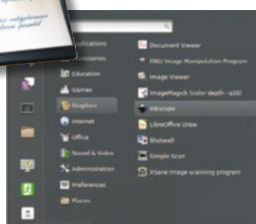
Stick/HDD'. The distro defaults to en_US language and keyboard, but you can also change this from the first screen.

The bottom panel on the Cinnamon desktop features the launcher on the left. There's also a dock at the top of the desktop, which by default has icons for *Nemo* file manager, *Firefox* and *HexChat* IRC client. Once you install Siduction, the dock also provides quick launch for *Image Viewer*.

We couldn't access the official forums as a Guest user, and couldn't register a new account either, but the comprehensive manual is a great resource for all users.

When trying Siduction we were prepared to run into frequent crashes, unresponsive tools, and other buggy behaviour. Instead, we found a fast distro that features an impressive array of default programs. **LXF**

Features at a glance



Bundled software

Siduction is filled with several useful apps across different categories, such as internet and multimedia.



Custom tools

Along with a customised Calamares installer, the distro features custom tools such as `simple-paste`.

LINUX FORMAT Verdict

Siduction 2018.1.0 Cinnamon

Developer: Siduction Team
Web: www.siduction.org
Licence: Various

Features	9/10
Performance	8/10
Ease of use	9/10
Documentation	7/10

» It's not intended to be run as a production machine, but on a VM it's ideal for testing new software.

Rating 8/10

Daphile 17.09

Do you own plenty of music, but are struggling to set up a headless music server? Then **Shashank Sharma** might have a solution...

In brief...

» The distribution is designed to help you store and play your music files from a server, which you can control from any machine on the network using a web browser. The music files must be hosted on the server itself, which makes Daphile different from music streamers such as Airsonic.

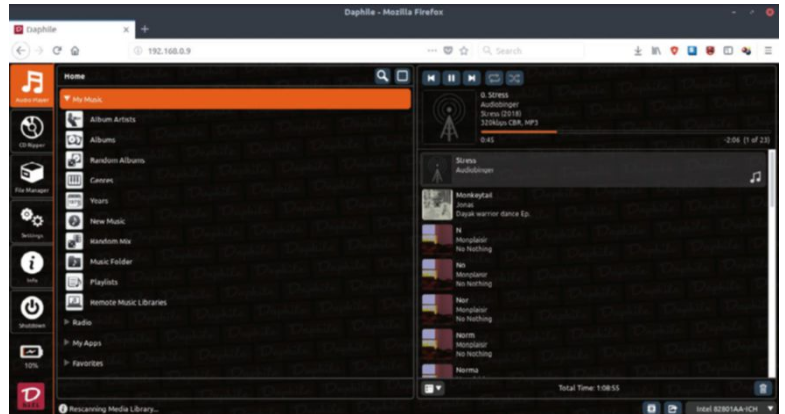
Daphile identifies itself as the heart of a digital music system. It's designed to help you play your collection of music files, rip CDs and even tap into radio stations, all from the comfort and convenience of a web browser. It isn't, however, a conventional music streamer.

Another feature that makes Daphile convenient to use is that it doesn't require installation. You can burn a CD or use `dd` to create a Live USB of the 200MB ISO and that's it. The distribution ships with all the tools and codecs required to help you play the music files that are already stored on the local hard disk.

The biggest drawback for the distro, however, is its licensing model. While there's no mention of it in the FAQ, the wiki or anywhere on the official website, a brief post by the sole developer on the DiyAudio forums reveals that the distro is "mostly based on open source (about 200 different packages). Everything that I have contributed by myself is proprietary and closed source".

Still more surprisingly, another post reveals that, "The source code is not distributed separately through Daphile web page. Daphile is build using mostly the standard Gentoo ebuilds. If you want the package list contact me through the Contact form in www.daphile.com and I'll send it to you. Some parts of Daphile that I have developed by myself are not open source".

We'll readily admit that its Gentoo origins, coupled with the small size, and



» Unfortunately, the project doesn't support ARM devices and there's no word on whether these would be supported in future releases.

zero setup initially caught our fancy, but its haphazard licensing approach might not sit well with some users.

Webby awards

If Daphile manages to detect and configure your network card, then the distro boots into a screen that shows the Daphile logo, along with a message that reads "Ready. Use the Web Interface. IP address: 192.168.0.8." There's no means of accessing any element of the distro, except to use the web interface.

The interface defaults to the Audio Player, which is the first tab on the sidebar. There are tabs for File Managers, Settings, CD Ripper and more. The two-pane Audio Player lists all available music, be it media files, radio stations and so on in a pane on the left. The file or station you play is displayed on the right pane. You can use the search bar at the top of the left pane to look for files. It supports searching by file, artist name, tags, and various other elements, but couldn't locate any files during our tests.

When running the Live environment, you can't change any of the default settings as Daphile reports that it's running in a read-only mode. This means that you can't configure the system to access Music files on other machines on your network using the CIFS, SSH, NFS or FTP protocols.

If you want the ability to tweak Daphile's settings, you have to install it

to disk. But for this, you must have a spare disk with no other partitions on it. In the web interface, click the Settings button on the sidebar and then scroll down to System Firmware and choose the disk to install it to. Unlike other Linux distros, Daphile doesn't let you add/remove software even after installation. Once installed, you can configure the music directories on different machines from the Settings>Networking menu. If all goes well, the internal devices, as well as network shares will be accessible from the file manager.

Daphile requires neither installation, nor any configuration before you put it to use, and can run even on 256MB RAM machines. If you want to turn an old machine into a music server, then Daphile is for you. **LXF**

Features at a glance



Browser interface

Despite the range of elements that comprise a full-featured music system, Daphile's interface is clean.

Multi-talented

Play music across different devices, rip CDs and enjoy radio across genres from different countries.

LINUX Verdict

Daphile 17.09

Developer: Kimmo Taskinen

Web: www.daphile.com/

Licence: Various

Features	8/10
Performance	8/10
Ease of use	7/10
Documentation	4/10

» Daphile provides an easy interface for a music server, and achieves this task without much fuss or bother.

Rating 6/10



The home of technology

techradar.com

Thimbleweed Park

Dead bodies, spooky spectral spirits, a verbal word salad of choice... **Andy Kelly** thinks it's just another normal day in *Linux Format* Towers.

Specs

» **OS:** Steam OS/
Ubuntu 64-bit
» **CPU:** 2GHz
» **Mem:** 4GB
» **GPU:** Intel HD
3000 or better
» **HDD:** 1GB

Federal agents Ray and Reyes are investigating a murder in the remote rural town of

Thimbleweed Park. A body was found under a bridge on the edge of town, but none of the locals seem to know anything about it or who the victim is. The agents have a checklist of tasks they have to complete to crack the case, including identifying the body and finding the murder weapon. But fingering out the killer won't be easy, because this is a *Monkey Island*-style point-and-click adventure game with a classic verb buffet interface: use, give, pick up, push and so on.

One of the first puzzles is taking a photo of the body, which is an introduction to the concept of switching characters and swapping items between them. But this gentle start soon gives way to the complex, elaborate puzzle chains the genre is famous (or perhaps infamous) for.

If you've ever wondered if it was possible to solve a homicide with a chainsaw, some sticky tape, and a coin, you're about to find out. Untangling a single problem can span several hours in *Thimbleweed Park*, and all of these objects are small pieces on opposite ends of a huge, elaborate jigsaw puzzle.

There are other playable characters: Ransome is a foul-mouthed clown living a lonely life, Delores is a game designer and meek pillow salesman Franklin is a ghost trapped in the hotel where he was killed.

Each character has a lengthy to-do list in their inventory, which gives you



» No, it's not Scully and Mulder, although the resemblance is noticeable.

direction without being too explicit. There's no hint system, but speaking to people and inspecting items in your inventory is often enough to steer you towards a solution. The game strikes a delicate balance between giving you subtle clues and steadfastly refusing to help you in any way, which makes solving a particularly tricky puzzle enormously satisfying. You feel like you've earned every victory.

The great outdoors

You will get stuck, but the game's open structure mean you don't encounter any brick walls. You can explore the town and the surrounding county (including the hotel and circus) freely, eventually unlocking a map that lets you travel between locations almost instantly.

With five playable characters, four of whom have inventories stuffed with items, including many that don't actually do anything, the game can occasionally be overwhelming. Sometimes we felt like we were fumbling around in the dark, hopelessly combining random items and trying every possible verb on every object we could find. But we always managed to claw our way out of those puzzle holes eventually, and the satisfaction of doing so ultimately made all the head-scratching and swearing worth it.

Thimbleweed Park avoids the 'adventure game logic' pitfall, possibly as the LucasArts '90s money-making hint line has long closed. It never feels

like it's being deliberately obscure just to make your life needlessly difficult.

Humour is often the glue that holds everything together in adventure games, and while *Thimbleweed Park* is funny, it does lack some of the warmth and charm of *Monkey Island*. It's overly self-referential and relies a little too much on sarcasm over actual jokes. But a cast of weird, colourful characters, entertaining dialogue, and a compelling central mystery keeps you interested.

Thimbleweed Park captures the essence of classics adventures while avoiding some of the things that made them frustrating. More genuine character interaction would have been good and sometimes the story gets a little too meta, but this is one of the best modern point-and-click adventures around. **LXF**



» Franklin may or may not assist your investigation.

LINUX FORMAT Verdict

Thimbleweed Park

Developer: Terrible Toybox
Web: www.thimbleweedpark.com
Price: £15

Gameplay	9/10
Graphics	9/10
Longevity	8/10
Value	9/10

» A must-buy quality adventure game with challenging puzzles, oddball characters, and an intriguing plot.

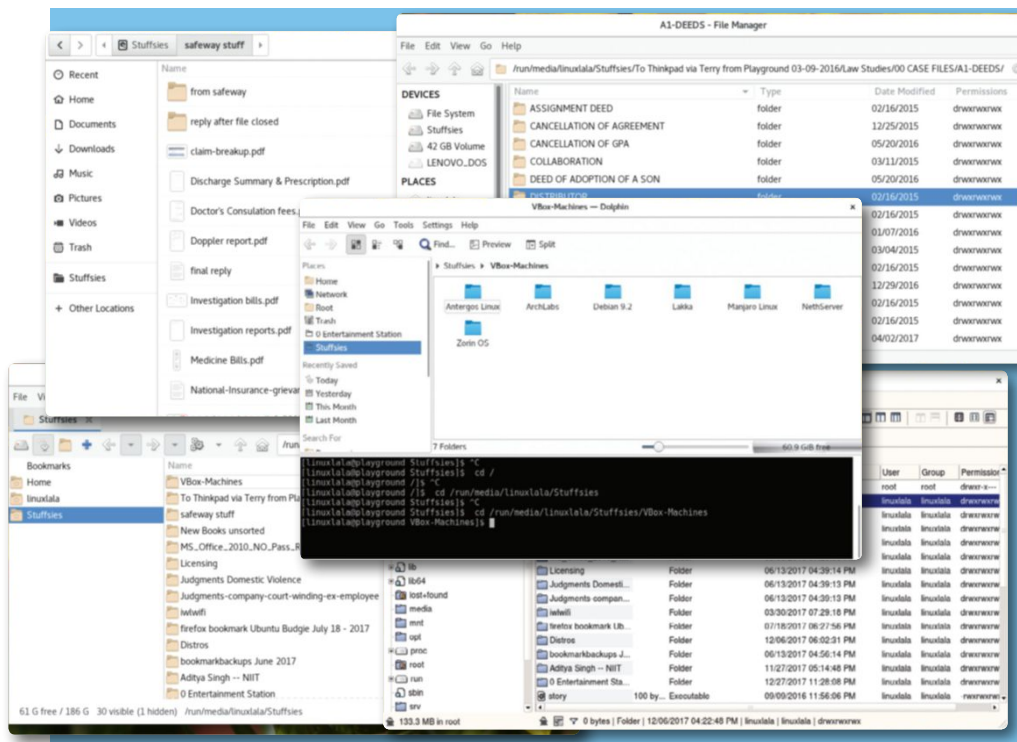
Rating 9/10

Roundup

» Every month we compare tons of stuff so you don't have to!

File managers

A file manager can do much more than simply help you navigate directories. **Shashank Sharma** checks out tools to help you make sense of your data.



How we tested...

The tools on our list were installed on a Fedora 27 Workstation. Starting with the Gnome edition, we installed the KDE environment on top of it to access the full features of Dolphin file manager. The other tools are all available from the software repositories of Fedora, and various other distros as well.

While the file managers themselves are lightweight, they sometimes require immense resources. Apart from performance, we'll be looking at the search capabilities on offer with these tools. We're also looking at how well the tools integrate with the desktop and let you perform encryption and compression operations.

We're looking for a tool that's easy to use. Bonus points if it's customisable and enables you to add functionality through plugins and extensions.

For a large section of users, a file manager is the nifty little tool you use to navigate the directories on your disk drive. Having been in existence for several decades now, the ubiquitous tool is now a key component of most desktops.

KDE, Gnome, Mate, Cinnamon, XFCE and even newer desktop environments such as Deepin each feature their own file manager. These tools are well integrated into the desktop and provide assorted functionality such as a search feature, the ability to create compressed archives, and more.

All the tools in our list have been in development for a long time, and with the exception of *SpaceFM* and *XFE*, are the default file manager in a popular desktop distribution (distro). While the default file manager should be sufficient for your computing needs, it's our sincere hope that this *Roundup* will reveal something fresh and interesting about the projects covered

and make you want to try them for their unique offerings.

To be fair, there's a large selection of file managers that we couldn't cover because of space constraints, and the self-imposed limitation of using only GUI tools. That said, you'll find a list of useful alternatives, including powerful command-line variants, in the 'Also consider...' section on page 29.

Our selection

- » Dolphin
- » Gnome Files
- » SpaceFM
- » Thunar
- » XFE

“It's our sincere hope that this Roundup will reveal something fresh about the projects covered”

Useful functions

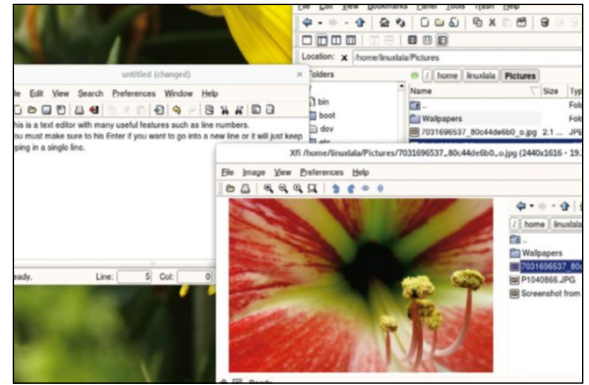
What makes them special?

The tools on our list can all perform the most basic function of navigating directories. All of them also support the breadcrumb feature, which helps you keep track of the directories as you traverse deep into a nested directory. *Dolphin* and *Gnome Files*, the oldest projects on our review list, also support the Undo and Redo features. While this feature has been a part of *Dolphin* for a long time, *Gnome Files* has only introduced this feature recently.

Gnome Files remembers the last operation performed and makes it possible for you to either undo or redo the same, as applicable. For instance, if you create a new folder and then create another new folder, it will enable you to undo the last action. You can undo the last created folder, which will delete the folder. If you, however, create a new folder, or copy some files into the new folder, then the undo option means you can revert the copy action.

Dolphin, on the other hand, possesses much greater powers of recall and remembers all your actions for the currently open window. The default file manager for KDE also features an integrated terminal. It uses Konsole as the terminal emulator so you must install KDE to access this feature. *Dolphin* also makes it possible to add tags to your files and folders. Although it's been in the works for some time, at the time of writing *Gnome Files* doesn't yet offer the tags feature, and it isn't even on the to-do list for the other files managers featured in this month's *Roundup*.

SpaceFM's Path Bar (location bar) is located in each panel above the file list for the current directory. At its simplest, the Path Bar enables you to see the current folder's path, but it also provides the breadcrumb feature. This means that you can Ctrl+click a portion of the path to switch to that directory. If you're in the `/home/linuxlala/`



➤ The lightweight XFE file manager comes with a complement of apps such as a text editor (xft), package manager (xpf) and image viewer (xfi).

Documents/articles/2017/PDF/ directory and Ctrl+click the Documents part of the path bar, then you'll immediately be taken to the `~/Documents` directory. You can also run commands from the path bar, without launching a *terminal* first, but you must use one of the following prefixes: `$` to run a task, `&` to run the command in background, `+` to run the command in terminal and, finally, `!` to run the command as root.

Verdict

Dolphin

★★★★★

Gnome Files

★★★★★

SpaceFM

★★★★★

Thunar

★★★☆☆

XFE

★★★☆☆

» Although full of features, Thunar and XFE come across as rudimentary in comparison to the others.

Desktop integration

Do these file managers play nice with other programs?

File managers come in different shapes and sizes but thankfully, they're no longer so intricately tied to their native desktop environment that you can't install any of the ones on our list on top of your current desktop environment. That said, it would be unfair to judge these projects harshly if a feature doesn't

work as advertised outside of their native environment.

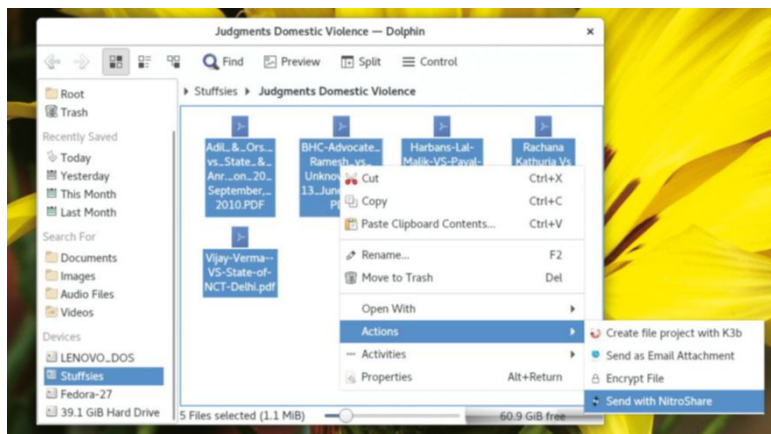
Both *Gnome Files* and *Thunar* enable you to set a selected image as the wallpaper from the right-click context menu. On our Fedora 27 Gnome installation, *Gnome Files* performed this task flawlessly, but *Thunar* could not. You can also create

compressed archives of selected files or send them as an email attachment with *Gnome Files*, but nothing else.

Out of the box, *Dolphin* is highly integrated into the desktop, even on top of Gnome, but only if you've installed KDE. This is because some of its features, such as create file project with K3b, Send as mail and encrypt rely on KDE-centric programs such as *Kmail* and *Kpgp* respectively.

All the projects recognise many different file types and provide appropriate actions such as extracting files from an archive, installing/uninstalling rpm packages, mounting ISO images, and so on.

Apart for *XFE*, all this month's projects make it possible to select files and send them as an email attachment, which requires the default email client to be configured. While it supports the option to email files, *SpaceFM*'s implementation of the feature is flaky – offering the option for only some file types, but not all.



➤ Thunar is the only tool that's unable to create a compressed file archive.

Verdict

Dolphin

★★★★★

Gnome Files

★★★★★

SpaceFM

★★★★★

Thunar

★★★☆☆

XFE

★★★☆☆

» Unlike the other projects, XFE doesn't automatically know what to do with ISO or PDF files.

Usability

User friendliness is a virtue.

A file manager's raison d'être is to help you make sense of your files and directories. This is why its organisational skills are so important. All of the tools on our list enable you to view the files in a directory as a list or with big and small icons.

You can also zoom in and out and drag and drop files. We also want a tool that makes it easy to switch directories as well as access removable media and other partitions on the disk. It's also good if the tools can create a bookmark for frequently accessed directories.

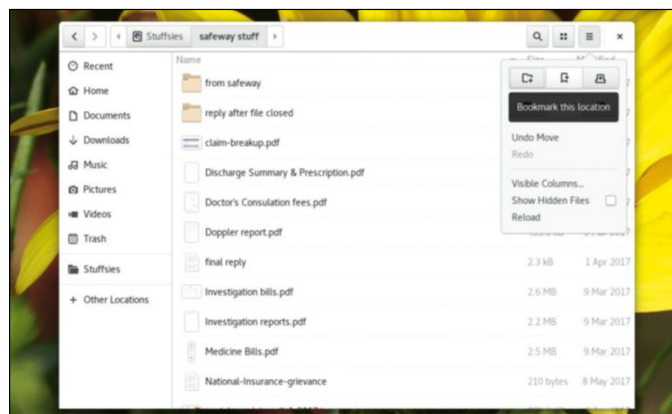
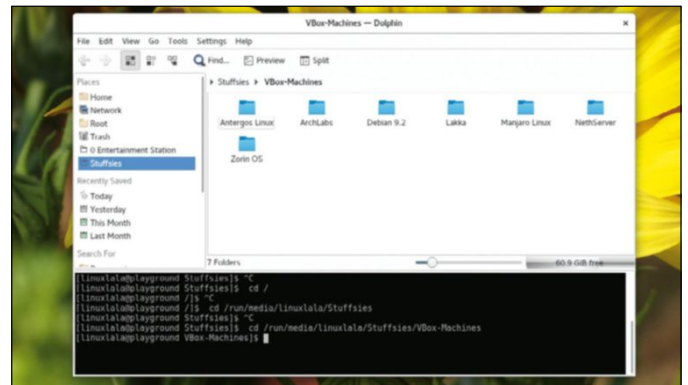
As your files grow, you may not always be able to recall the contents of a file merely from the filename. The same is also true for images, because a descriptive file name may not be enough for you to remember what the image is, and so the preview feature is also important.

Dolphin ★★★★★

Entries on the sidebar are split into Places, Recently Saved, Search for and Devices. You can click any item, and *Dolphin* will list all the files you've modified since first boot of the day. You can also configure *Dolphin* to list additional information for the files and folders in the current directory.

From the Preferences window, you can also select the services you want available in the context menu such as Run in Konsole or Send via Bluetooth, and *Dolphin* features a large number of options out of the box.

If your current directory has a host of image files, you can click the Preview button on the top bar to view thumbnails. You can drag any directory to the top of the left sidebar to create a bookmark, or right-click the Places heading on the sidebar and click Add entry.



Gnome Files ★★★★★

Unlike *Dolphin*, *Gnome Files* features a minimalist interface with neither toolbars nor too many buttons cluttering the workflow. You can create new folders and zoom in or out. You can also bookmark the current directory from here or simply drag-drop a directory onto the left sidebar to create a bookmark.

By default *Gnome Files* shows the file name, size and modified date for each file and folder in the current directory. Click the Visible Columns button if you need additional information such as Type, Owner or Permission. You must click the Files icon in the panel on the desktop and then click Preferences if you want to configure the tool.

If you want to open files/directories with a single click, you can configure it from the Behavior tab on the Preferences dialog. Like *Dolphin*, *Gnome Files* enables you to open multiple tabs in the window.

Documentation & support

Help is always welcome.

A part from an overview of its features, the KDE UserBase Wiki also provides hints, tips and various tutorials on how best to use *Dolphin*. These cover file system navigation, using custom icons for different folders, using the bulk rename feature, and more. The File Management page on the wiki discusses at length the interface, panels, bookmarks and other features. Although there isn't a dedicated forum board, you can use the official KDE forums should you need any

assistance. The Handbook, which you can access by pressing F1 from *Dolphin*, is another useful resource.

The *Gnome Files* help discusses all of its features and is split into different categories such as common tasks, file related tasks, removable drives and even features some tips.

In addition to a user-contributed Wiki hosted on GitHub, *SpaceFM*'s user manual on the website provides a thorough introduction to the tool. The user manual also discusses in detail the

different configuration options and the functionality provided by the Panels, Path Bar and so on. User-contributed plugins are listed on the project's wiki.

The forums hosted on SourceForge are an additional resource and you can also connect with users and developers on the #spacefm IRC channel.

Accompanied with screenshots, the documentation for *Thunar* covers various aspects of the file manager such as the interface, working with files/folders and useful plugins.

Verdict

Dolphin ★★★★★
Gnome Files ★★★★★
SpaceFM ★★★★★
Thunar ★★★★★
XFE ★★★★★

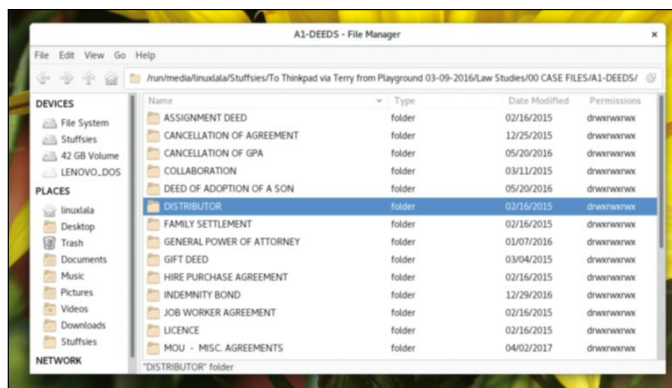
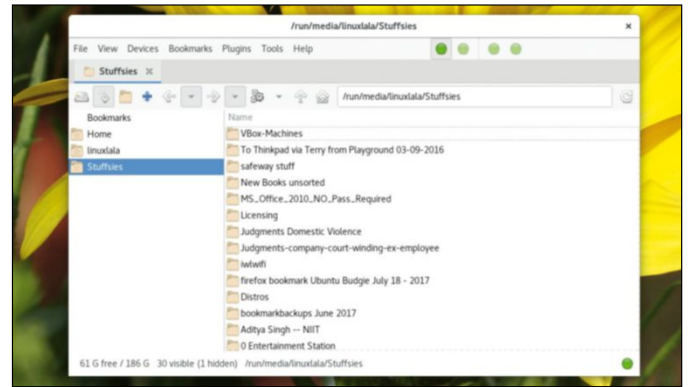
» The XFE website provides an introduction to its features, but nothing else.

SpaceFM ★★★★★

Unlike *Gnome Files* that supports tabs but not split panels, *SpaceFM* enables you to open four panels and you can open several tabs in each panel. This is useful if you work with several directories at the same time, since each panel can be used to independently navigate the filesystem. For each panel, you can also choose to have a tree of the file system, bookmarks, and a list of removable devices displayed on the sidebar.

The tool remembers the last accessed directory and opens new instances with the same directory, unlike the other tools which default to the `~/` directory. *SpaceFM*'s interface is also highly customisable.

While the rest of the tools in this *Roundup* support zoom, which makes it possible for you increase the size of the icons in the current directory, neither *SpaceFM* nor *XFE* provide this functionality.

**Thunar** ★★★★★

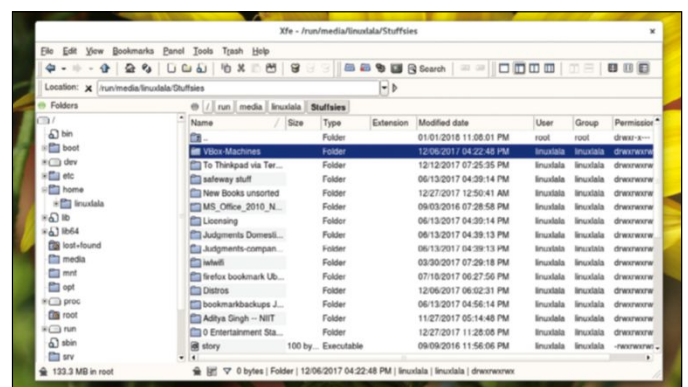
The interface presents a single status bar, menu bar, and a side pane that can either show bookmarks (called shortcuts) or filesystem tree. From the Preferences window you can configure the tool to open files and directories. You can click View>Configure Columns if you want more than filename and size displayed in the main panel.

Along with *Dolphin* and *Gnome Files*, *Thunar* is the only other tool that supports bulk rename and shares another feature with *Gnome Files*. Both these tools support the use of Templates. This is a document formatted to your specifications, which you can use to create new documents. You can create a template that has the necessary formatting and the bare bones structure. You can then right-click and select Create Document> <document-template> to create a new file.

XFE ★★★★★

Unlike the other tools on our list, *XFE*'s default interface features a number of toolbars adorning a large number of buttons. The tool supports launching a terminal (Xterm) out of the box. If you use an alternate terminal, you must change the configured apps from the Preferences window. *XFE* similarly enables you to launch *XFE* as root user, by pressing the Shift+F3 key combo, but it didn't work for us.

The main window is referred to as a panel and *XFE* offers several interfaces. In addition to the default view that presents a tree list on the sidebar and a panel, you also have the choice of a two-panels interface, which resembles the split interface on *Dolphin*. The other options are 'one panel' and 'two panels with tree'. Unlike the other tools, *XFE* doesn't show preview or thumbnails of image files.



Supported protocols

Can these file managers connect with remote machines?

To help you connect with a variety of machines on your network running assorted operating systems as well as remote machines, these file managers also support different protocols. This means you don't have to install additional graphical programs to access remote machines.

Most of the file managers on our list support different network protocols. The functionality, however, isn't provided by these tools themselves, and you have to install the underlying

components yourself. For instance, you must have the requisite packages for Samba already installed if you wish to access Windows machines on your network. Also, *SpaceFM* and *Thunar* use *udev* and *gvfs* to provide support for the various protocols so make sure these are installed on your machines, before testing these file managers.

Apart from using Samba to connect with remote machines, you can also use these file managers as FTP clients to access remote servers. If you've

installed the necessary NFS components, you can also mount and then access directories on another Linux machine on the network from the comfort of your favourite file manager. For users who still prefer to use SSH and FISH protocols to securely transfer files to remote machines, the same can also be done using these file managers.

XFE is the only dud in this test. It doesn't support any protocol and can only be used to browse the local disk and partitions.

Verdict

Dolphin
★★★★★
Gnome Files
★★★★★
SpaceFM
★★★★★
Thunar
★★★★★
XFE
★★★★★

» The tools are evenly matched and performed well when moving files.

Plugins and extensions

Who says no to more features?

Dolphin doesn't provide a centralised list of supported plugins or extensions. The KDE store, however, does provide an easy access to *Dolphin's* Service Menus from under the KDE App-Addons heading on the sidebar. Each of these add further options to the context menu, such as the convert jpg to png addon, which is self-explanatory, and

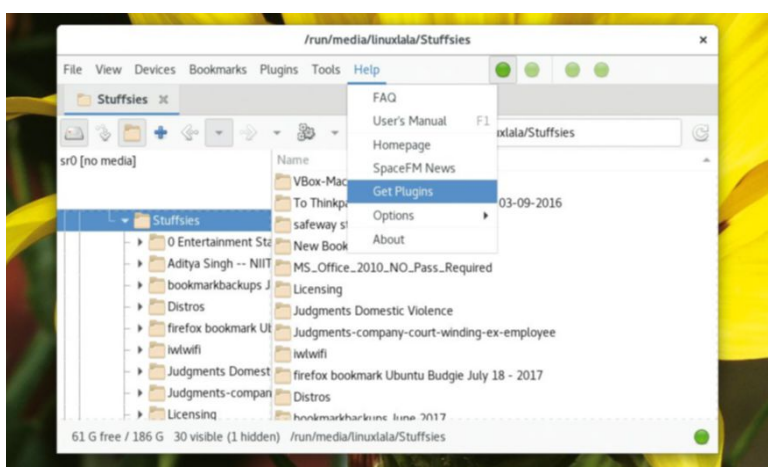
uses *Imagemagick* to convert the images. There are similar extensions to convert media files to different formats, export *Libreoffice* documents to PDF, and so on.

As with *Dolphin*, *Gnome Files* also doesn't feature a list of plugins or extension on the project's website or the wiki. Since the project is still referred to by its original name

internally, run the `dnf search nautilus` command on Fedora for a list of plugins available in the software repositories. You can then install useful extension such as *gnome-terminal-nautilus* and *nautilus-image-converter*. There are similar extensions that help you tie Gnome Files with Dropbox, ownCloud and other services.

The *SpaceFM* wiki lists all available plugins such as the ClamAV plugin which makes it possible for you to scan the selected files/folders with ClamAV. There's a similar plugin to help you encrypt/decrypt the selected files/folders with GPG. Creating plugins for *SpaceFM* is quite easy and the entire process is discussed at length in the official documentation.

You can find a handful of plugins for *Thunar* on XFCE's official Git repositories. The archive plugin adds the option to create archives from the context menu. The shares plugin similarly gives users the option to quickly share folders using Samba. The media tags plugin add ID3/OGG support to the bulk rename dialog.



» You can access all the plugins for *SpaceFM* by clicking Help>Get plugins.

Verdict

Dolphin
★★★★★
Gnome Files
★★★★★
SpaceFM
★★★★★
Thunar
★★★★★
XFE
★★★★★

» *XFE* lags in this test – it doesn't provide plugins to extend its functionality.

Search flexibility

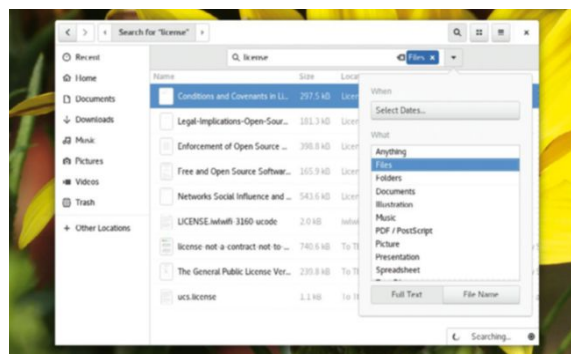
It's a file manager's make or break feature.

The two sides to a coin metaphor perfectly describes the ever-increasing disk space sizes. On the one hand high-capacity disks make it possible to retain files on your system rather than relegating them to a DVD backup. The downside is that you often have to wade through a large number of directories when searching for a file. This makes knowing your file manager's search capabilities so important.

Dolphin relies on Baloo, the file indexing and search framework for KDE Plasma to handle all search queries. This means that you must let Baloo keep an active index of all the files on your system. If you don't, the search will return zero results even when you know a particular file definitely exists on disk. Apart from wildcards, you can also search for files based on type, such as audio or document, and narrow the results based on rating.

With *Gnome Files*, you can restrict search to files based on type. Apart from PDF, picture, spreadsheet, presentation, video and other types, the tool also enables you to select from an exhaustive alphabetised list of formats. You can also define the last modified or last used date. For searching within the current directory, the tool also supports pattern matching. This means that you can search for files using patterns which describe the file name, such as `*trip*.png` which will select all png files within the current directory that have the work trip anywhere in the filename.

Search on *SpaceFM* is relegated to a separate window which you can access by clicking File>File Search. Although it supports a lot of pattern matching variables discussed in the manual, the search is flaky and unpredictable – it works sometimes, but not always.



» Unlike the other tools, *Gnome Files* automatically performs a recursive search and is much more adept than its peers at finding matches.

XFE can search within hidden files. You can also restrict the search based on size, type or the last modified date. You can even search for files using the permissions: for example, you can limit the search to all files created by the *linuxlala* user and last modified before 60 days with the permission 0644. You can also specify if you're looking for a folder, file, socket, link or pipe. The search can also be recursive and you can set the tool to follow symbolic links.

Verdict

Dolphin
★★★★★
Gnome Files
★★★★★
SpaceFM
★★★★★
Thunar
★★★★★
XFE
★★★★★

» Like *Gnome Files*, *Thunar* also supports pattern matching, but that's it.

File managers

The verdict

On paper, all the tools in this *Roundup* are fairly well matched. *XFE*, despite being a single developer project does well to hold its own against mighty projects supported by a robust team of dedicated developer and vast user communities. Not only does *XFE* lack any plugins or extensions, it also doesn't support any network protocols. This lack of support for SMB, NFS and other protocols is unfortunate and so *XFE* comes in last, despite its robust search capabilities.

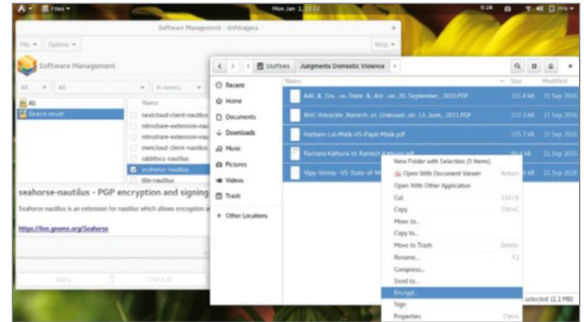
Unfortunately, *Thunar* has almost no search skills to speak of, and is thus out of the podium race. Despite taking the last two spots, *Thunar* and *XFE* are not without merit. If you're unlikely to use a file manager to connect with remote machines, you'll be pleased with *XFE*'s simplicity and performance.

Even though *SpaceFM* doesn't yet support undo/redo operations, it ranks higher than *Gnome Files* in the useful

feature test because of its robust path bar, which can be used to run commands without first launching a separate terminal. Still, its unreliable search feature and inability to provide proper context menu options for all file types have pushed it into last place.

Dolphin and *Gnome Files* were fairly evenly matched in all tests, except desktop integration where *Dolphin* outshines the others out of the box. But once you install a few *Gnome Files* extensions, you'll get all the same functionality offered by *Dolphin*.

In addition, the undo/redo feature of *Gnome Files* is of limited use because it only remembers the last operation, unlike *Dolphin*, which remembers all operations for the currently open window. Hopefully, the developers will work on this feature and adopt a more *Dolphin*-esque approach to undo/



➤ Many distros, such as Fedora and Ubuntu, provide a number of useful extensions for Gnome Files.

redo operations. Although *Gnome Files* doesn't enable users to add tags, the implementation on *Dolphin* seems to be of limited utility, since it doesn't permit the use of tags as a search criteria.

The most important factor in deciding the winner of this *Roundup* was the search feature. *Gnome Files* wins the top slot because of the speed with which it churns out matches for the search terms.

“Not only does XFE lack plugins or extensions, it also doesn't support any network protocols”

1st**Gnome Files** ★★★★★

Web: <http://bit.ly/gnome-files> Licence: GPLv3+ Version: 3.26.0

» Simple interface and impressive features net this file manager first place.

4th**Thunar** ★★★☆☆

Web: <http://bit.ly/thunar-fm> Licence: GPL Version: 1.6.13

» Recommended for those who swear they won't ever search for files.

2nd**Dolphin** ★★★★★

Web: <https://dolphin.kde.org> Licence: GPL Version: 17.08.1

» A faster search would easily result in *Dolphin* winning the contest.

5th**XFE** ★☆☆☆☆

Web: <http://roland65.free.fr/xfe/> Licence: GPL Version: 1.42

» Ideal if you just want a file manager that can also search for files.

3rd**SpaceFM** ★★★★☆

Web: <http://bit.ly/space-fm> Licence: GPLv3+ Version: 1.0.5

» The well-designed and highly configurable tool could use more testers.

Over to you...

Upset with our ranking or peeved at us for ignoring your favourite file manager? Email your opinions to lx.f.letters@futurenet.com.

Also consider...

There's no dearth of file managers for Linux distros. We've already run *Roundups* discussing some of these alternatives in **LXF86**, **LXF119** and more recently (*7 years ago!*-Ed) in **LXF143**. The tools discussed in this *Roundup* belong to the category of navigational file managers. The more traditional file managers, such as the ones covered in **LXF119**, are

classified as orthodox file managers. If you have no qualms working with the command-line variants, try *Midnight Commander*. It's an excellent and robust tool and offers several popular features such as bulk rename.

If you favour either the Mate or Cinnamon desktop environment, these feature their own file managers – *Caja* and *Nemo* respectively –

and are comparable to *Gnome Files*. *Rox Filer* is another alternative for the Rox desktop.

The Gentoo file manager, which incidentally pre-dates the Linux distro of the same, is another robust and desktop-neutral choice, but only if you don't mind spending copious amount of time configuring it to your own particular requirements. **LXF**

Subscribe to

LINUX FORMAT

Get into Linux today!

Choose the perfect package for you!

» GET THE PRINT EDITION



» Every issue comes with a 4GB DVD packed full of the hottest distros, apps, games and loads more!

Only £18

Every 3 months by Direct Debit

» GET THE DIGITAL EDITION



» The cheapest way to get *Linux Format*. Instant access on your iPad, iPhone and Android device.

Only £11.25

Every 3 months by Direct Debit

» GET THE BUNDLE DEAL

Get both the print & digital editions for one low price!

£24

Every 3 months by Direct Debit

**SAVE
36%**



» PLUS: Exclusive access¹ to the *Linux Format* subs area—1,000s of DRM-free issues, tutorials, features and reviews.

Subscribe online today...

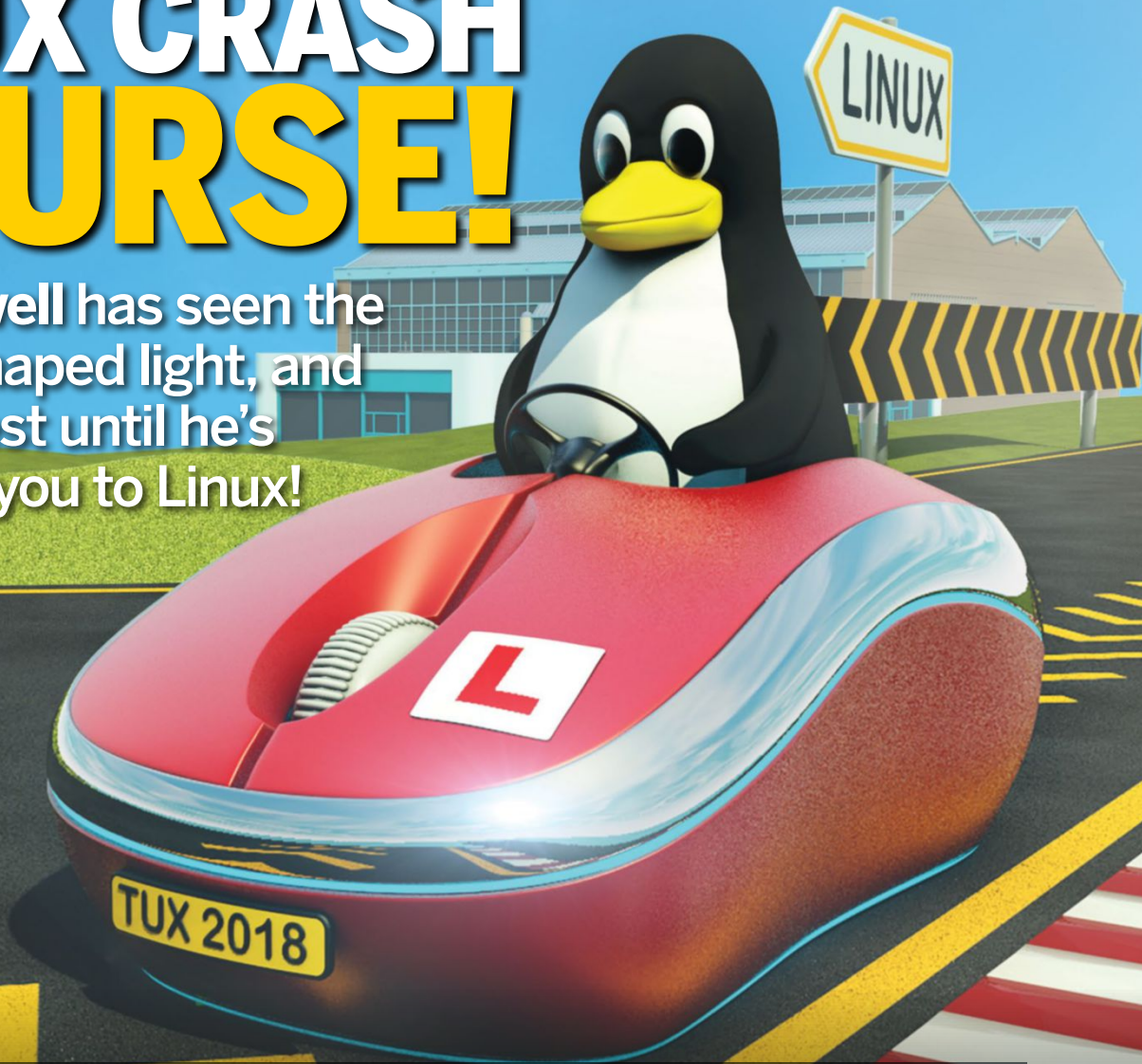
www.myfavouritemagazines.co.uk/subLIN

Or telephone: **0344 848 2852**

Prices and savings quoted are compared to buying full-priced UK print and digital issues. You'll receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. Your subscription is for the minimum term specified and will expire at the end of the current term. Payment is non-refundable after the 14-day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at time of print and subject to change. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) and are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: <http://bit.ly/magtandc>. Offer ends 31 March 2018. 1) Only with subscriptions taken out via 'MFM'.

60-MINUTE LINUX CRASH COURSE!

Jonni Bidwell has seen the penguin-shaped light, and he won't rest until he's converted you to Linux!



Psssst. Have you heard about Linux? It's a free operating system that's just as capable as anything else on the computing market. But not only is it free in the sense of it having no price tag, it's free in the sense that it can be moulded into anything your imagination (or coding prowess) can conceive and free in terms of no one tracking you, pushing you adverts or unwanted software.

This freedom not only covers the operating system, but it extends across the whole software ecosystem surrounding it. Free, Libre and Open Source Software (FLOSS) is the polar opposite to the walled-garden approach proffered by commercial

desktop OSES, where we are only allowed to install things from app stores, where new privacy-eroding features are introduced on a daily basis and where control is subtly wrested away from the user under the guises of 'ease of use'.

“All of Linux's source code can be modified by anyone who's willing to put in the time”

There are literally thousands of free, libre applications that are every bit as good, and in many cases better than their commercial counterparts. All the while you won't have to forego access to the latest hardware or even lose access to your

gaming collection. It is easy to become overwhelmed though, even if Linux has never been easier to use. Learning new ways of working is tricky, and sometimes being experienced in one OS turns out to be a serious downside when moving to another. One's first Linux steps are often wracked with confusion and uncertainty. They are often followed by shouting, wailing and gnashing of dentures. If one is not careful, one

may even end up stuck in something called a *Vim*. Shocking.

We'll save all of that fun for later, and let us help you set up and learn the beautiful, powerful Elementary OS – one of the most exciting Linux distributions on the scene.

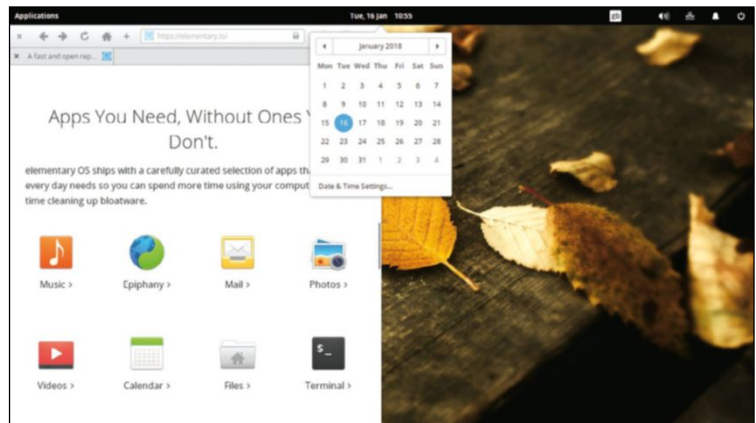
It's a brave GNU world...

We help you get your bearings around the terra incognita that is GNU/Linux.

Since you've read past the intro it seems reasonable to assume you've some interest in installing and dabbling with Linux. Bravo. You may have heard great things about software freedom, about fighting back against software monopolies, or about the awesome power and configurability of Linux. By the same token, you may also have heard gripes about document incompatibility, rants about simple tasks being impossible without recourse to arcane terminal incantations, and terrifying accounts of being stuck in *Vim* (a text editor) for days. There's truth in all of these. Linux has a lot to offer, but operating systems and users both can be fickle creatures, and sparks can fly (sometimes literally) when they clash.

See the box below for a quick primer on what Linux is, or read the Wikipedia entry. It's good to know where we came from and what free software is all about. It's common for people to conflate "Linux" with "desktop Linux distribution" which causes all kinds of confusion. Linux in general is used everywhere from tiny embedded systems to the most powerful supercomputers in the world. The desktop is probably the only area it's failed to dominate.

Today's desktop Linux distros feature slick installers, impressive GUIs, repositories full of the latest applications, and package managers for easily installing them. These things are all decouple-able, which can be a hard concept for refugees from macOS and Windows to grasp. For example, the idea that you can install a new desktop environment without affecting underlying system settings and then continue to use all your applications as normal, might seem a bit alien. Indeed, the whole package management system is entirely at odds with downloading and running some random executable file from the publisher's website (or a dubious mirror), but more on that later. You also won't believe the things that can be done from the command line.



We'd be fools if we pretended installing Linux was rainbows and unicorns though. Things do go wrong, trainwreck wrong sometimes. For example, the 17.10 release of Ubuntu had to be pulled because it was corrupting the UEFI settings of certain Lenovo laptops, rendering them unbootable. And if you look in the Installation & Upgrades section of the Ubuntu forums you'll find at least one post per day from someone whose installation attempt has failed in some mysterious way. There's usually more than meets the eye to such stories though, and for every calamitous installation there are probably a hundred that go without a hitch. People are less inclined to post when things go swimmingly, which is a shame really.

You can try out Linux straight from the *Linux Format* DVD without so much as touching any other OSes you may have installed. This won't be as slick as running a proper install, and you won't be able to save any changes you make, but it's a great and hassle-free way to start exploring.

» This lovely desktop can be yours in not much longer than it takes to make a cup of tea.

»

What is GNU/Linux?

Volumes could be spent on this topic and still not provide a satisfactory answer. But if you want to play with Linux it's useful to have a handle on what it is and the philosophy that supports it. So here goes (stop reading if journalistic oversimplifications annoy you).

The free software movement started in earnest with Richard Stallman's GNU Project, which ultimately sought to grant users the freedom to use, share, study and modify the software running on their computers. To this end, they set out to create a UNIX-like OS called GNU (a recursive acronym for GNU's Not Unix). Many of the popular UNIX tools were ported to GNU. The GNU project also formalised their ideals in a license, the GNU GPL (General Public License), which included the all important "copyleft" provision, meaning that derivative

works must be released under the same license. New software was written too, such as the GNU C Compiler and GNU Emacs, which are sine non qua for today's Linux distros and text editor arguments respectively.

Despite this progress, what wasn't finished (and to this day remains unfinished, though not abandoned) was the kernel. The kernel is the hardcore bit of the OS that talks to the hardware, manages memory and basically handles all the complexities your average user never has to think about. Writing a kernel is hard, but in 1991 a young student named Linus Torvalds took to Usenet to announce he had done just that. By combining his "Linux" kernel and porting the Gnu Tools to it, a new OS was born. It was licensed under the GPL in 1992 and the first "distributions" (back then these were

just the kernel bundled with some software and documentation and crude installation mechanism) soon followed.



» GNU's mascot is a Gnu, who Gnu?

Installing Elementary OS

Be the envy of your friends with one of the finest distros around.

» **W**e often get some stick for being too Ubuntu-centric. It's a fair point – there are hundreds of other distributions. It's just that (at least as far as desktop Linux is concerned) most people are using Ubuntu. Last issue we strayed from the trodden track a little by going with Linux Mint for our Build a PC feature.

Mint is an Ubuntu-derivative that differs from its progenitor by its use of the more traditional Cinnamon desktop. Under the hood, things are much the same. This time around, we're going to use Elementary OS, another Ubuntu-derivative with another desktop. Both Elementary

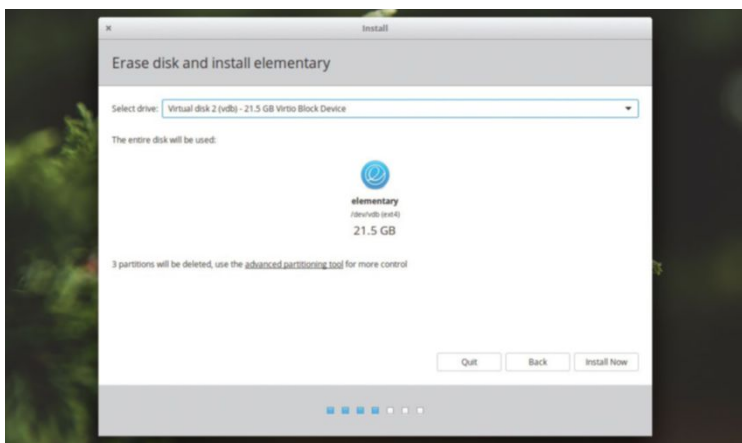
and Mint are based on Ubuntu 16.04 LTS, so they inherit all of its stability and receive all of its prompt security updates.

Elementary is technically still in beta, but don't let that put you off. We've been following its development avidly over the past couple of years and have been nothing but impressed. Of course, Linux is all about choice, so feel free to choose Ubuntu or Mint, or whatever distribution takes your fancy. Some of this guide will be specific to Elementary, but much of it applies to any Ubuntu-based distro, and the additional tools mentioned later on are available for any flavour of Linux.

Elementary's Pantheon desktop is often described as MacOS-like, which in some circles is a compliment, but we think it deserves more credit. Granted, the launcher bar at the bottom and the multipane file manager do bear more than a passing resemblance to the fruity OS, but it's more than capable of standing on its own merits. Apart from looking pretty, Elementary has some unique features that make it worthy of your attention. We'll delve into those more once we've got the thing installed.

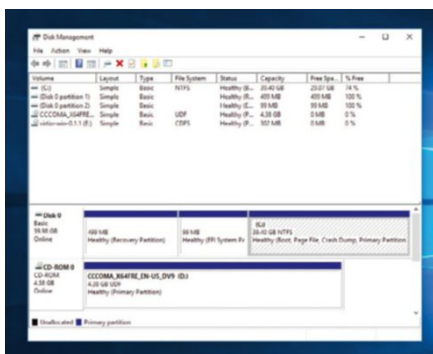
Duelling OSes

Now is an excellent time to back up any important files you have on your machine. If you're setting up a dual-boot arrangement with, say, Windows, then it's unlikely that installing Linux will do bad things to your Windows partitions. It's also not unheard of though, so don't take the risk if you rely on the target machine for mission-critical business. Consider what would happen if your Windows install breaks. Do you have the means to re-install? Do you have license keys



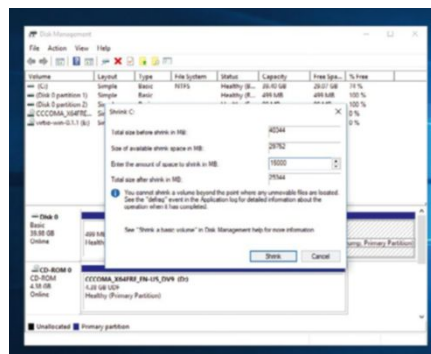
» If you want to install Elementary OS its own drive, select the Erase disk option. You can choose which disk to erase on this screen.

Making room for Linux



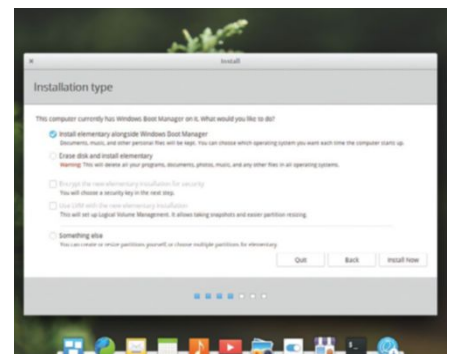
1 Disk management

Boot into Windows (*this seems an odd way to start a Linux feature – Ed*). After the slew of updates, notifications and forced restarts (*ooh burn – Ed*) open the Start Menu, start typing **disk** and select Create and format hard disk partitions. This will open the *Disk Management* console where you can investigate the partition structure of any drives that are installed.



2 Shrink existing partition

It's important not to touch either the EFI or any recovery partitions – doing so could be catastrophic. Select the partition where Windows is installed (probably C:), right-click it and select Shrink Volume... You'll need at least 10GB to install Elementary, but we'd recommend at least 30GB if you can spare it, or more if you need it. Adjust the amount and then press Shrink.



3 New partition

Follow the text above for installing Linux up until the Installation type screen. Choose the Install alongside Windows Boot Manager option. If you're on a modern UEFI system then you may need to need to change settings to boot the **ubuntu** entry rather than Windows. GRUB (elementary's bootloader) menu will have an option to boot Windows, should you ever need to return there.

for it and all your other programs to hand? If you have a spare SSD or hard drive then consider using that for your first install, sidestepping any potential territory war. Better yet, if you have a spare computer, even an old one, then consider using that for your first install.

Having carefully heeded our warnings and deciding you still want to proceed with dual boot, then follow the step-by-step guide (*below left*). Note that those steps are only necessary if Linux is going on the same drive as Windows. If not, you can install straight from the *Linux Format* DVD.

Support network

Check out the DVD pages and FAQ (<https://linuxformat.com/dvdsupport>) if you have difficulties getting the disc working. You'll need to ensure Secure Boot is turned off for it to boot, and you may need to tell your BIOS/UEFI to boot from the DVD rather than its hard drive. This is beyond the scope of this feature, but the FAQ has some hints. If your target machine doesn't have an optical drive then you can download an ISO from <https://elementary.io> (the donation is optional) and follow the instructions at <https://elementary.io/docs/installation> to write it to a USB device.

For the first successful boot choose "Try elementary OS without installing". The annotation below will help you get your bearings once it's booted. Explore the installed applications, don't be put off by any slowness, then, when you're ready, select Install elementary OS from the Applications menu.

The installation will ask you a couple of questions along the way. If you're connected to the Internet, then you'll want to check the "Download updates while installing..." box. The main reason for ticking the "Install third-party software" box would be to make your wireless card work, so if you're not

sure it's best to check this box, too. Only select the "Erase disk and install elementary" if you're sure there's nothing on the target drive. There's a confirmation screen before anything destructive happens, so use that to make sure the correct drive will be used.

You'll then be asked for some localisation info, and then to set up a user account and password. It's a good idea to tick the "Encrypt my home folder" box if you plan on storing sensitive information on this machine. This will prevent the data being read if the hard drive is removed, but won't protect against malware accessing it while you're logged in. It also means your data will be as good as lost if you forget your password. So don't do that. Instead make a cup of tea, wait patiently for the installation to finish. Hit Restart Now, remove the installation medium when instructed to do so and press Enter. If things go wrong you may need to brave the UEFI

"If you have a spare computer, even an old one, then consider using that for your first install"

settings. Many implementations are buggy and forgetful, but it should be possible to manually boot from here.

All being well you should, after a few moments gazing at the elementary OS logo, be met with a login screen. Enter the credentials from before and you should find yourself in an environment remarkably similar to both the live environment we booted to initially and the annotation below. Have a browse around, then continue reading for more top tips and guidance on how to make the most of elementary OS. »

Know your way around Elementary

Applications

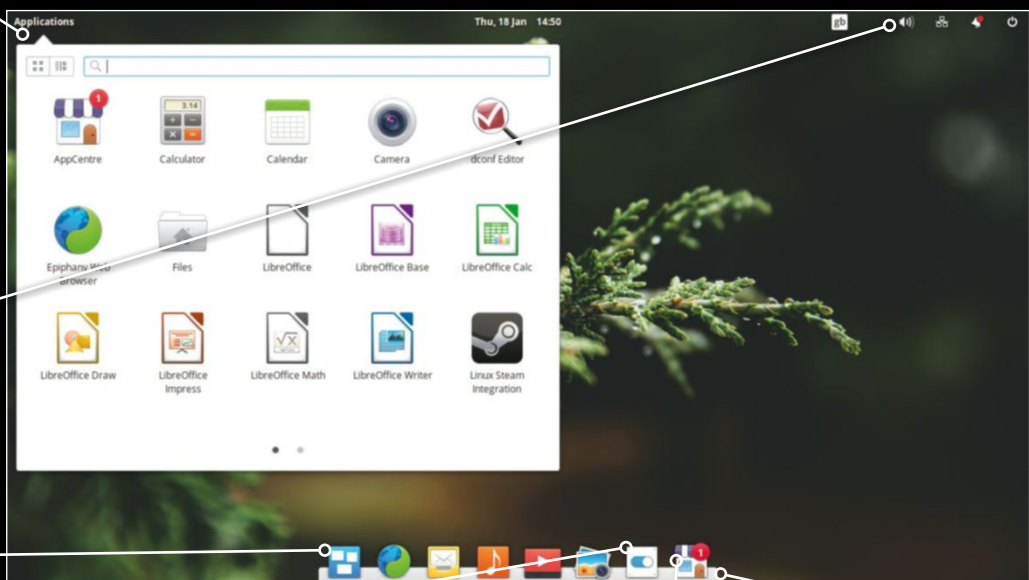
All the graphical apps are to be found here. You can use the Windows-Space keyboard shortcut (Alt-F2 also works) to save your mouse pointer a trip. Some users change the shortcut to just the Windows key, as is the norm in Windows, Unity, Gnome and KDE.

Indicators

Here you can access sound and network controls, notifications and shutdown options. The notifications area will alert you about System Updates, imminent calendar appointments and more.

Multitasking View

Open a couple of applications. Click this button, then add a new desktop and open other apps there. This can improve your workflow by designating certain computing activities to certain desktops.



Settings

Control all conceivable system settings from one handy place. Default applications, languages, parental controls and more can all be taken care of from here.

AppCentre

This is the easiest way to add or remove applications to Elementary OS. Click the icon to launch it, and you'll find a selection of applications, including some specifically curated for Elementary OS.

Plank Launcher

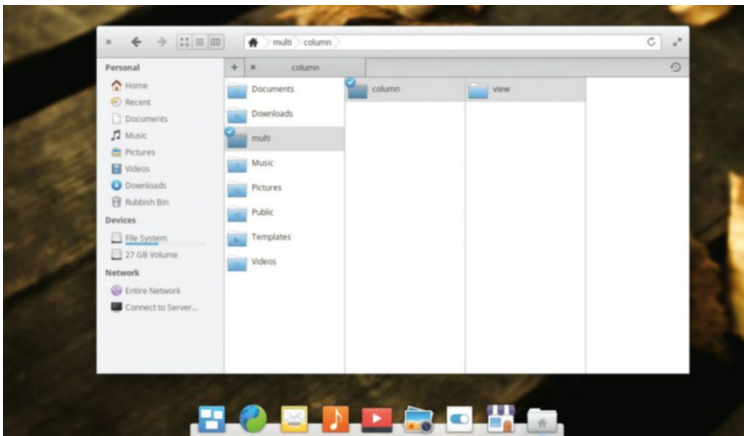
Plank houses shortcuts to commonly used applications, and it also works as a dock for currently running applications (denoted by a blue dot beneath the icon).

Exploring Elementary OS

Peruse the plenitudes of programs for work, play and edification

» **O**ne thing you won't find on Linux is bloatware. On Windows this is generally added by manufacturers and OEMs, or as unnecessary extras that come with device drivers. Those few manufacturers that do supply Linux on their machines know better than to commit such an atrocity, and because most drivers are included in the Linux kernel itself they don't have need of these unwanted extras.

Even compared to other Linux distributions, the number of applications bundled with Elementary OS is pretty small. Most of those included in the default install are unique to Elementary, so integrate nicely with the minimalist feel of the Pantheon desktop. Don't be put off by their lack of menus and options. These apps, particularly the file manager, have been engineered to provide powerful features without the need for complex configuration. *Scratch*, the text editor, (which will be renamed to *Code* in the next release), remembers your tabs and autosaves documents. Even ones that haven't been named yet.



» **Activate the multicolumn view with the button just to the left of the location bar for more efficient transacting of inter-folder business.**

Distros that come crammed to the hilt with applications can be confusing, especially for beginners. Better to start out with the basics (some people are happy with just a web browser and a music player) and add what you need as you need it. If your life revolves around (or at least involves frequent dealing with) documents, spreadsheets and presentations, then you'll need an office suite. There are a few options on Linux, but the most popular is the excellent *LibreOffice*. We could install this from the AppCentre (although there seems to be a long-standing issue which makes this harder than it should be, see <http://bit.ly/libreoffice-wont-open>), but now is as good a time as any to introduce the command line.

From the applications menu click Terminal. You'll be greeted with an expectant and vaguely intimidating-looking flashing cursor. This is where advanced Linux users love to do their work. Once you get used to it, this becomes an incredibly powerful way of working. You can play music, check your email or write scripts to automate dull tasks. Actually pretty much everything you can do from the GUI can be done at the command line. We'll use it to install *LibreOffice*, which is achieved by running:

```
$ sudo apt install libreoffice
```

SuperUsers last all summer long

Before we talk about the scary output, let's dissect the above command. The **sudo** part indicates that what follows is to be run as root (similar to the Administrator account in other OSes), it's short for (SuperUser-do). Whenever you see a command beginning with **sudo**, make sure you understand what it does, since it has the potential to do harm to your system. Running commands as a regular user can only do harm to that user's files.

Apt is the package management suite used by Debian-based distros. Software in Linux is distributed as packages that are housed on repositories (repos) maintained by distribution teams. Packagers take the software from

There's life yet in 32-bit computing

One of Linux's oft-touted advantages is that it runs on old hardware. This is true, but there are limits. For example, the Linux kernel dropped support for Intel's 386 processor (the very architecture on which Torvalds' first Linux builds ran) at the end of 2012.

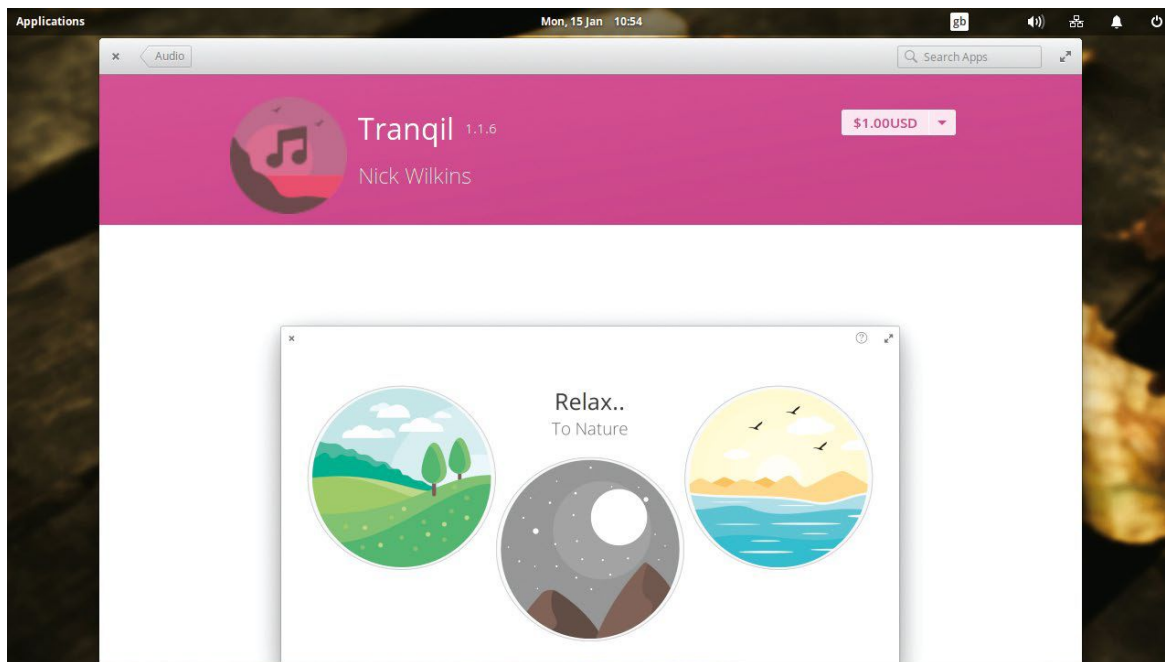
Breathing life into old hardware is a nice idea, but it's easy to underestimate the demands that "regular desktop use" imposes on such machines. Most distros gave up supporting x86 CPUs lacking PAE memory management (introduced with the Pentium Pro in 1995), even though the kernel still supports them. So while using a 486 with Linux isn't impossible, you'll have your work cut out.

Lately, distros have started to discuss winding down support for 32-bit x86 altogether. It's already happened in Arch Linux (though a community maintained fork is alive and well). Fedora now only provides a minimal 32-bit install media and Ubuntu desktop will be doing the same for the upcoming 18.04 release.

Although demand is waning some specialist distributions will continue to support 32-bit x86 installations. One such is the excellent Legacy edition of Bodhi Linux, which is also on the *Linux Format* DVD. Like elementary this is based on Ubuntu 16.04, so despite the obvious cosmetic differences the installation procedure and system layout are all the same.



» **Under the hood the lightweight Bodhi is identical to Elementary OS.**



» You'll find bespoke Elementary OS apps, such as Tranquil [sic] in the AppCentre. You don't have to pay, but calm is priceless.

upstream (the application developers themselves) and tailor it so that it plays nicely with their distribution. New technologies like Snaps and Flatpaks (see page 46) do things differently and enable application developers to distribute distro-agnostic packages, but let's not worry about that right now (we'll touch on Snaps later). Packages are signed and checksummed for security and integrity, and can easily be removed without reliance on some shoddy uninstall utility. The rest of the command is pretty self-explanatory: we want *Apt* to **install** the **libreoffice** package.

So now, onto that scary output. *LibreOffice* is a big bit of software, and depends on many other things (a Java runtime, fonts, various nondescript libraries) also shipped as packages and known as dependencies. The individual *LibreOffice* components also have their own packages, named **libreoffice-writer**, **libreoffice-calc** and so forth. *Apt* is asking us to confirm that we do indeed want to download and install all of this gubbins, and telling us how much space will be taken up by doing so (about 450MB when we tried).

Since we live for clerical duties, let's go ahead with the install by hitting enter (the capital **Y** indicates it's the default, you can type **y** and hit Enter if you want, too). After a few coloured ASCII-rendered progress bars the process will finish, and if you check out the Applications menu you'll find shortcuts to the various facades of the *LibreOffice* suite. Firing up *Writer* may bring back memories of *Microsoft Word* circa 2000, but if you look beyond such trivialities you'll find a word processor more than capable of prosecuting all your character and paragraph-related business.

Back to the Appcentre

Although installing software from the *AppCentre* seems vastly different to installing packages from the command line, behind the scenes that's exactly what *AppCentre* does. With very few exceptions, almost everything that you can do with a mouse and a GUI app can be done from the command line. For example, elementary OS will periodically check for updates and (very politely) notify you if any are pending.

What's really happening here is that the command (or at least one functionally identical to it):

```
$ sudo apt update
```

is being run in the background. This updates *Apt*'s cache of available packages. Once that's done any updates can be applied with the following:

```
$ sudo apt upgrade
```

As you explore Linux more, you'll find lots of examples beginning with these incantations. If the package cache is stale, then you'll run into errors when *Apt* starts trying to fetch outdated packages that no longer exist in the repos. Our long-running and ongoing Terminal series (available to subscribers from the *Linux Format* Archive) will provide more

“Terminal enables you to play music, check your email or write scripts to automate dull tasks”

in-depth coverage of the command line, but for now we'll just mention one more command: **man**. This is short for manual, and is exactly what the oft-iterated acronym RTFM encourages you to read. For example, if you want to know the ins and outs of the **ls** command for listing the contents of a directory (what the vulgar refer to as a “folder”).

Elementary ships with the *Epiphany* browser from the Gnome desktop (which is now just known as Web). This is more than capable, and is powered by WebKitGTK so it conforms to all the latest web standards. Be that as it may, you may prefer to use *Firefox* (it now supports Netflix and other streaming services out of the box), which you'll find in the AppCentre, or *Google Chrome*, which you won't. You will find Chromium there though, which is the open source version of *Chrome* that lacks the creepy bits and the restrictive redistribution policy. If you really need the full-fat *Chrome*, you'll find installation instructions at www.google.com/chrome/browser/desktop/index.html.

»

Getting hold of help

Every pro geek was once a newbie, so don't be afraid to be baffled.

» **T**hings do go wrong, and in general the Linux community is friendly and willing to help. But there's a lot you can do to help them help you. First, remember this is free software. You didn't pay for round-the-clock support, so be grateful if someone tries to help you. Rants of the variety "Linux isn't ready for primetime" are generally not welcome or useful.

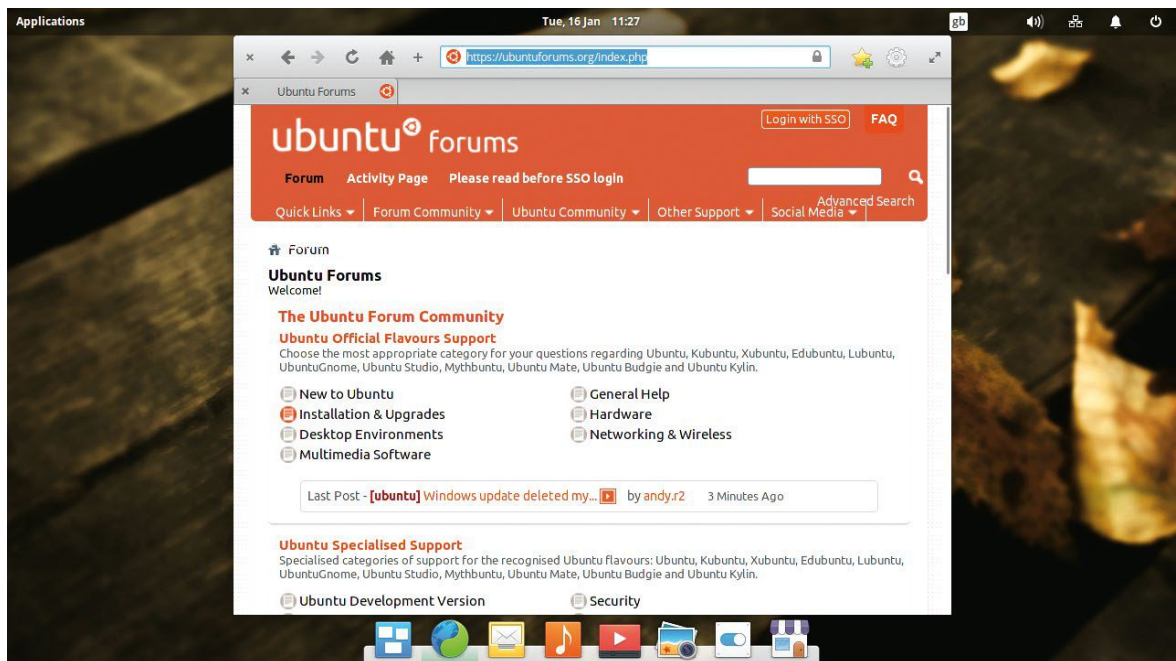
That may be your opinion or experience, but hundreds of thousands of happy desktop Linux users probably say otherwise. Penning those rants and then deleting them is a good way to calm down (and not feel so foolish the next morning). Part of the joy of Linux is solving problems, although in the beginning it's hard to know where to start. For Elementary OS, its support site (<https://elementary.io/support>) is a good start.

Conveniently brushing aside installation problems, your first challenge might be to get wireless working (you may be one of the lucky ones with a chipset that works off the bat, in which case spare a thought for the less fortunate). Most of the time wireless woes stem from the fact that the code which runs on the wireless radio chips is proprietary and can't be redistributed. This doesn't mean it's in any way illegal for you to obtain it yourself.

Solve your wireless woes

The first step is to find out which chipset we're dealing with. The make and model number aren't necessarily the best guides here, since manufacturers have a habit of changing hardware between revisions. The Terminal `lspci` command tells us all about any PCI devices connected to our system

» The Ubuntu Forums over at <https://ubuntuforums.org> are a great place to find answers. Both Bodhi and elementary use Ubuntu under the hood.



Bugs vs “you’re using it wrong”

Linux behaves strangely sometimes, so strangely in fact that you might have cause to wonder if that behaviour is intended. For example, you might wonder why you can't use the mouse to position the cursor in the terminal-based *Nano* text editor, or why the usual copy and paste keys (Ctrl+C, and so on) don't work there.

These examples are most definitely intended. There is no mouse input (at least not from the desktop) for console applications, and Ctrl+C serves the very important function of

sending the SIGINT (user interrupt) signal to console programs (which causes them to gracefully close). You might also dislike that you can't put icons on Elementary's desktop, or even right-click it (well you can, it just doesn't do anything). That's by design too.

Learning to distinguish these quirks from unintended behaviour is just a matter of experience. There are lots of occasions where you'll need to adjust either your perspective or some obscure setting when you are precluded from achieving some goal.

Bugs do creep in though, and one of the joys of open source software is that processes for wrangling them are transparent. We'd advise holding off reporting bugs until you're au fait with the processes involved (collecting the right bits of logs, working with patches, compiling code, generating backtraces). But do have a look at the Ubuntu bugtracker to see how these things go. For example, check the details of the Ubuntu 17.10 firmware mentioned earlier at <https://bugs.launchpad.net/ubuntu/+source/linux/+bug/1734147>.

(there's `lsusb` for USB devices). That command generates lots of output, so we'll filter it using `grep`, so we only hear about network devices. Open the Terminal and run:

```
lspci -nnk | grep -iA2 net
```

The `-nn` part tells `lspci` to give vendor and device codes, which look like `1814:3090` and uniquely identify PCI hardware. The `k` tells it to give information about any kernel drivers that can handle the device. We "pipe" (with the `|` symbol) the output to `grep`, to restrict it to entries only containing the text `net`. Use `man grep` to find out more about `grep`'s many switches. You'll probably see information for both your wired and wireless interfaces, but it should be reasonably clear which is which. In our case, the relevant output was

```
02:00.0 Network controller [0280]: Ralink corp. RT3090
Wireless 802.11n 1T/1R PCIe [1814:3090]
```

```
Subsystem: Lite-On Communications Inc RT3090
Wireless 802.11n 1T/1R PCIe [11ad:6622]
```

```
Kernel driver in use: rt2800pci
```

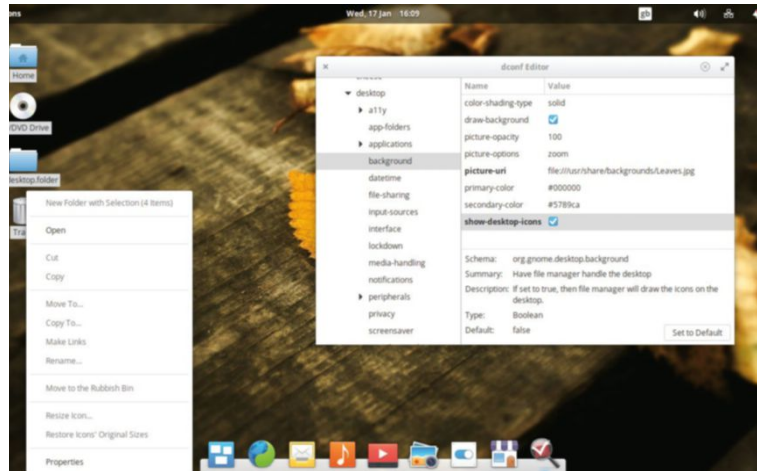
This card works out of the box, but if it didn't we'd be diligently searching (on the Ubuntu and the Elementary fora), including in our search terms the driver (`rt2800pci`) and the device codes (`1814:3090`). A lack of available firmware won't stop the driver being loaded, (though it will stop it doing anything useful). If the output for your wireless device doesn't report a kernel driver in use, it's possible you have a different problem. Hopefully one that the above investigation helps you to solve, a good place to start is the Wireless hardware section of the Ubuntu Wiki at <https://help.ubuntu.com/community/HardwareSupportComponentsWirelessNetworkCardsRealTek>.

Find help... fast

Recalcitrant wireless drivers are far from the only thing that impedes people's first steps with Linux, but whatever your problem there are a few general things you should do before seeking community assistance. Linux may not be as popular as other desktop operating systems, but it's popular enough that someone else has already run into the same issue.

First, spend some time searching forums and the web for the particular error message or the hardware that may be causing it. Second, don't blindly go copying and pasting terminal commands from random corners of the Internet. Error messages can be red herrings: the same one may turn out to be catastrophic in one situation but harmless in another. Trying to solve a problem that doesn't exist instead of the one at hand generally results in two or more problems. So don't follow someone else's solution unless you're sure you're on the same page and check release versions.

Do read support threads in their entirety. There's unfortunately evolved a special class of forum poster who will post all kinds of convoluted 'solutions' to a problem. Sometimes this is well-meaning, and sometimes the given solutions may even work, but often it's just someone wanting to show off their "mad" Linux "skillz" and belittle or confuse the original poster. Said skillz turn out not to pay the bills when, for example, the proffered solution breaks down after a system update. Unless you're trying to do something hideously complicated, or something went hideously wrong, the solution you seek ought to be fairly straightforward. Check the date on any posts you come across. A solution from 2003 is unlikely to still apply today.



» **Elementary's lack of option for enabling icons on desktop isn't a bug, but seek and a solution ye shall find**

Unfortunately, some of the most common beginner problems are the scary ones: the system dumping you at the GRUB rescue shell, or flashing up some text quicker than any human can read and then dumping you at a black screen.

Super Grub2 Disc (www.supergrubdisk.org) is a rescue disc that might be able to help with bootloader troubles. The dreaded black screen is usually indicative of misbehaving graphics drivers (almost always the Nvidia proprietary drivers, but sometimes old or quirky hardware is to blame). If you're lucky you should be able to login by switching to a console (as opposed to the graphical display manager) by pressing `Ctrl+Alt+F1`. From here you should study the `X.org` file with `less /var/log/Xorg.0.log` for any clues (scroll through it with the Up and Down arrows, and press `Q` to quit).

“Don't blindly go copying and pasting terminal commands from random parts of the Internet”

More generally, you can look at the whole system journal for the current boot with `journalctl -b`. If something went wrong and you had to reboot, then you can see messages from the previous boot with `journalctl -b -1`. The system log receives all kind of information: some of it makes for interesting reading, but most of it is the computer equivalent of parochial chatter. Journal entries are stratified into eight levels, numbered zero to seven (programmers like to start enumerating things at zero, much to the chagrin of normal people), where zero is the most severe ("emergency") and seven the least ("debug"; programmers like to enumerate things backwards too). Most log entries above level three ("error") won't relate to things that went wrong.

To see only the more important messages use the `-p` (priority) switch, for example `journalctl -b -p 3`. Information you find there might at least help you or generous forum elves to localise the problem. Don't be afraid to ask for help, but be considerate when you do. This stickied post on the Ubuntu Forums covers the basics: <https://ubuntuforums.org/showthread.php?t=1422475>.

»

Do more with Elementary

Digging deeper to solve problems is a worthwhile endeavour...

» **W**e've hopefully given you a decent overview of the basics of Elementary OS, and shown a bit of how Linux in general works. Elementary perhaps goes further than any other distro to be completely usable without resorting to the command line. This makes it ideal for beginners, but it also hides a lot of functionality.

For example, there are some great things in the *AppCentre*, but they represent only a tiny fraction of what's available in the Ubuntu repositories. If they all were visible in the same layout, no one in their right mind would scroll all the way to the end, so we see only a curated selection. If you really want to see the whole package selection in a graphical application, then install *Synaptic*: it will show you the truth about packages in all their incomprehensibly named glory. But there's more to life than checking out Ubuntu's massive package selection. You may have searched hither and yon for an option to move the Close button from the left, or perhaps enable the Minimise button.

These and other seemingly immutable elementaryisms can be made malleable through the elementary *Tweaks* tool, which you won't find in the AppCentre.

Elementary, Dear Watson

One of the joys of open source software is that you can take someone else's code, tweak it slightly compile it yourself and package it for your distribution. One of the troubles is that this process can be frustrating and time-consuming, and often needs to be repeated whenever the tool or any of its dependencies are updated.

Ubuntu introduced the idea of Personal Package Archives (PPAs), which enable developers to host their own packages without having to undergo the stringent checks required to get them into the official repos. Users can choose to add a particular PPA, and thus sidestep all the hair-pulling traditionally associated with rolling your own packages. The only caveat is that you trust the PPA, since anyone could set one up and host all kinds of nasties. So be wary of adding these without doing some research first.

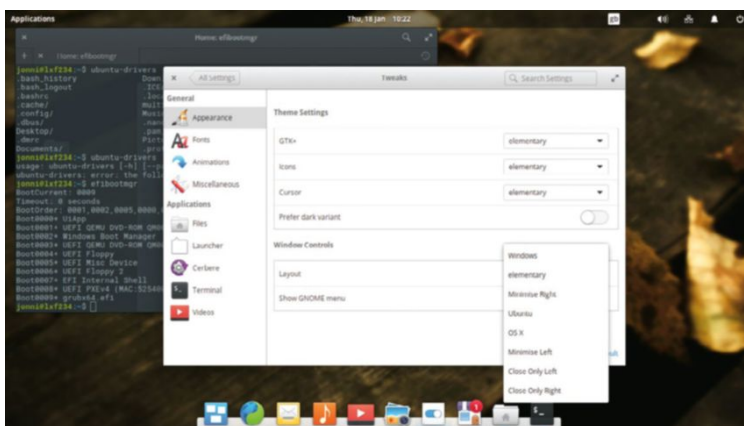
Fire up the Terminal application again and pass it this series of commands:

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:philip.scott/elementary-tweaks
sudo apt-get update
sudo apt-get install elementary-tweaks
```

Now open up System Settings, and lo and behold you'll find a new Tweaks applet, where window controls and other things can be rejigged to your heart's content.

As a reward for sticking with this feature to the end, we'll leave you with some tips to further your gaming experience with *Steam*. We hope you've enjoyed it, and that you enjoy Elementary OS. Follow the Elementary team's blog at

<https://medium.com/elementaryos> and send Jonni your Elementary questions at lxformat@futurenet.com. **LXF**



» **Elementary Tweaks** makes it possible to change toolkit and icon themes as well as put whichever window controls wherever you desire.

Installing Steam

The Linux gaming scene has grown considerably ever since *Steam* came to Linux in 2013. Most triple-A titles don't make it to Linux, and those that do generally don't perform as well as on Windows. Nonetheless, you'll still find thousands of Linux titles (including the excellent *Thimbleweed Park*) into which you can happily sink all of your free time.

You won't find *Steam* in the *AppCentre*, but it can be installed with a simple `sudo apt install steam` (use Tab and the cursors to negotiate the licensing dialogs). If you want to try something a bit adventurous though, it's worth complementing this with the

Steam Integration Tool from Solus OS. This stops *Steam* from using the old runtime libraries it ships, intercepting system calls and redirecting them to newer libraries (those provided in the first snap below). It also contains a number of fixes for popular titles, which could save some hair pulling.

```
$ sudo apt install snapd
$ sudo snap install --edge solus-runtime-gaming
$ sudo snap install --devmode --edge linux-steam-integration
```

If you run into difficulties it can be easily removed with the following:

```
$ sudo snap remove linux-steam-integration
```



» **Fans of Maniac Mansion, Monkey Island** can relive those memories with Ron Gilbert's *Thimbleweed Park*. (see p23)

YOUR ESSENTIAL GUIDE TO THE TECH YOU SIMPLY CAN'T LIVE WITHOUT

From the ultimate smart home setup and the best 4K tech on the market to essential fitness and lifestyle gadgets, the T3 Annual 2018 brings you the best of T3's features, head-to-heads, reviews and more

T3

ANNUAL 2018

YOUR ESSENTIAL GUIDE TO THE BEST GADGETS AND TECH

ON SALE
NOW



FUTURE

Ordering is easy. Go online at:

www.myfavouritemagazines.co.uk

Or get it from selected supermarkets & newsagents

ON HIS EARLY EXPOSURE TO LINUX

“The first distro that I tried was called MiniLinux. You could install it from MS-DOS”





Alberto Garcia is a developer for Igalia, a rather successful open source consultancy company. From humble beginnings in their native Galicia, in

northwestern Spain, Igalia has become a truly international organisation that are involved with all kinds of open source technologies, from browsers and web engines to – Alberto's own forté – virtualisation with KVM and QEMU. We caught up with him during October's 2017 Open Source Summit at the Prague Hilton.

Linux Format: How did you get into computers, Linux, programming and the related dark arts?

Alberto Garcia: I had an early interest in computers. I think I received my first one aged 11 as a Christmas present from my parents. Back then games consoles weren't so popular in Spain, so I got a Spectrum and learned how to program on it. Around that time there was a lot of technical stuff in Spectrum magazines, too: they weren't just about games, there was a lot of details about computer internals in them. So I learned a lot about that from typing out code in the magazine.

LXF: Betraying my age slightly, I too remember the days of PEEK and POKE

AG: Exactly. Those were the days. So by the time I was writing my first games and small programs I decided that I liked computers and wanted to work with them. So I studied computer science in 1996 and that's where I learned about Linux, too. It wasn't particularly popular back then.

LXF: I tried to get Slackware working around that time. I was unsuccessful.

AG: I think the first distro that I tried was called MiniLinux. You could install it from MS-DOS. It used this filesystem called `umsdos` that implemented POSIX on top of the FAT filesystem. So you could unpack a zip file and run the Linux kernel without having to repartition anything.

Soon after this I moved to Debian. It had the largest selection of software and you could install it from CDs. I liked Debian a lot. I liked being able to see all the details of how everything works under the hood. Back then I felt like I had pretty much mastered MS-DOS, that there were no more secrets there, so Linux became the new challenge. Throughout our studies we had to use UNIX a lot and I discovered I liked that too, so I figured it would be nice if I could actually work on this.

Most of my peers were, in their professional lives, working on Java, Windows NT or whatever boring stuff was popular back then. But I was more interested in Linux, so when I graduated »

Sound & Virtualisation

Jonni Bidwell heads to the Prague Hilton and asks **Alberto Garcia** everything you wanted to know about virtualisation but were afraid to ask.

» I, along with some classmates, founded a company. Our goal was to work on Linux and do open source in general, and that company was Igalia. We started working locally. The market was very different back then. Most of the things you see now didn't really exist, so it was a very different company. But we very quickly became involved with the Gnome project. Little by little, we started to work less with local customers and local development and more with upstream communities, contributing directly to those projects.

LXF: Is Debian still your distro of choice?

AG: It is actually. I've been a Debian maintainer for some years now, too. In fact, I think I've still got the version of Debian that I installed back in 1997, or whenever it was. I copied the data to a new hard drive and put in a new machine, and that copy has been kept up to date. But yeah, the original install is still there, too: there are still files from 20 years ago. I like Debian's centralised development model. There's very little hierarchy – I like how it's run. For me it was natural to start contributing to it. I started by maintaining a couple of packages, but now I'm a bit more involved. Our company maintains the WebKitGTK+ port and I package it for Debian. If there's a problem, Debian enables us to see how the package runs in different architectures and helps with the upstream development.

LXF: Tell us a little more about Igalia

AG: We're an open source consultancy company based in A Coruña in the Galicia region of Spain. We were founded in 2001 with two goals: open innovation and FLOSS development. In the beginning there were 10 of

us, but now we have 60 engineers, distributed across several countries.

More than half the company works with web browsers and associated technologies: we do some work with Mozilla, quite a lot with WebKit and, lately, a lot of Chromium too. We do a lot of graphics work, too. We work with Intel on Mesa development, as well as lots of other things related to toolkits and optimisation in general. We work with compilers and virtual machines, too. We're mostly focused on JavaScript, so we have a team working with Google on V8, a team working with Mozilla on SpiderMonkey, and a team working with Apple.

We also do lots of multimedia stuff, this month we were at the *Gstreamer* conference and hackfest, which was also in Prague. We maintain its WebKit backends and also contribute to core development as well. Those areas probably account for about 90 per cent of what we do, we also work on Accessibility, we contribute to the kernel and we have a small team working on a next-gen networking toolkit called *Snabb* (see www.igalia.com/networking). I'm involved with the virtualisation side of things, other people in this team work with related cloud technologies, such as the CEPH storage system.

Historically, we've worked on all sorts of different projects. We're in lots of different areas today, but I'd say the core one is web browser

engines. Ten years ago we started maintaining the WebKitGTK+ port, and most of the development there was done by us. We discovered that there was a lot of interest in having an open source web engine that manufacturers can use in their products.

Before the WebKit fork we were the one of the largest contributors, behind Apple and Google. So we grew in that area and started some new projects in closely related areas. Web engines are connected to areas of the whole stack, so we need people that know about multimedia, compilers (for the JavaScript interpreter) and so on. Nowadays, we have people working on the web core, implementing new standards and features; people working on the JavaScript compiler, making it faster; multimedia, people working on GPU acceleration. Aside from that we have a lot of experience working with open source communities and have other teams working in different areas.

LXF: It's hard to keep up with WebKit, WebKitGTK+, Blink and all the other web engines. Can you give us a bit of background about how WebKitGTK+ came about?

AG: I wasn't initially part of that team, but we were very involved with the Gnome desktop environment and Gnome had, and still has, a browser called *Epiphany*. This used the *Gecko* web engine, but as far as I remember, Gecko was not designed to be embedded. It was designed as a web engine for *Firefox*, so if you wanted to use it somewhere else it didn't

quite fit. WebKit on the other hand was designed to be just a web engine, and so was built with embeddability in mind. So it seemed like a good idea to add a GTK layer on top of it, so that we could use it in Gnome. Now it's used not just by Epiphany, but all kinds of other Gnome applications that need to display HTML or web content in general.

LXF: What is the focus of your work?

XX: I'm working with *QEMU*, particularly the disk I/O and storage side of things, so not really anything to do with web browsers. In the past I've worked on virtualisation of hardware, kernel device drivers and suchlike.

LXF: This is quite an action-packed conference: besides incorporating LinuxCon and ContainerCon, there's also MesosCon, the Kernel Summit and, most relevantly for you I guess, the KVM Forum.

AG: Yes it's great these things are co-located. I can be manning the booth, promoting the company. But I'll also give a KVM talk for the Forum later. It's a great opportunity to meet people from different communities.

ALBERTO'S DEBIAN INSTALL

"The original install is still there, too: there are still files from 20 years ago."





LXF: Can you give us a preview of your talk?

AG: Sure. So I'm working with the *QEMU* hypervisor. When you create a virtual machine you need to store its data somewhere, and *QEMU* has this native file format called QCOW2. QCOW2 has lots of features. You can create backing files, snapshots, encryption, compression and it grows on demand. So when you start with an empty device it takes up only a few hundred kilobytes, and when you add to it, it grows. This is nice, but it doesn't perform as well as a raw file, say a 20GB file that you access as a block device. A raw file offers the fastest possible performance here.

With QCOW2 there are some overheads. You need some data structures to handle the snapshots, the translations between the addresses the VM and host see with the file, and so on. My work is to reduce that overhead and make QCOW2 perform as fast as raw files. In many cases it does, but there are a few cases where *QEMU* still needs to be tuned, a few where the QCOW2 format itself needs to be extended, and a few where we need to improve the driver.

LXF: For someone used to VirtualBox, it can be hard to see how the kernel, KVM and QEMU all fit together. Can you provide some insight here?

AG: KVM (Kernel Virtual Machine) is the component in the Linux kernel that enables it to operate as a hypervisor. So it's a kernel module that means userspace programs can create virtual machines. KVM is independent of *QEMU*, so you can create a virtual machine using just the KVM API, specifying the number of cores, amount of memory and so on. KVM does all of that and gets the kernel to do all the low-level CPU scheduling. *QEMU* provides the emulation of all the input and output device drivers, and all the hardware. So basically KVM just creates the VM and isolates the call, and lets everything run there. *QEMU* provides emulation of disk devices, displays... all of these sorts of things.

LXF: I've been using Virt-manager for all my virtual machines for a while, and that leverages the libvirt API to manage VMs. Can

you explain where this fits into the scheme of things?

AG: So *QEMU* in its most basic form is a userspace program, it's a desktop program that you can launch on your own computer. Libvirt is a library that enables you to control virtual machines. You can create them, start them, stop them and destroy them. It's an API that works on top of *QEMU*, but it also works on top of all kinds of other virtualisation options. So you can have something above it that handles VMs, and libvirt provides an abstraction layer between the VM provider itself, and the application that manages everything.

LXF: Our readers will be familiar with the idea of snapshotting VMs, but tell us about the other advantages of QCOW2.

AG: Compression is pretty handy, so QCOW2 images are divided into clusters (similar to blocks in a filesystem) and you can compress those clusters. The resulting image is much smaller than what you would get with a raw file. There's an overhead involved because whenever you read a cluster you have to decompress it, but it's a trade-off. It also supports encryption; this year it got LUKS-compatible encryption. It also supports backing files that are closely related to snapshots.

So you have a base file with a base image and an active file where data is written. When you need to read data that isn't in the base file you go to the new file. In this way you can have several different VMs using the same guest OS and share common data between them.

LXF: What about resilience benefits? If I pull the plug while my QCOW2-backed VM is running, is my data still going to be there when I fire it up again?

AG: Well the QCOW2 format checks that every time all the writes to disk are done in a way that if the VM crashes the file isn't corrupt. But if it is corrupted, then at least we can detect it. So when we develop the format, we have to take into account the ordering of the writes and try to make sure data isn't lost. If data is lost then we have to make sure we can detect that.

LXF: QEMU has come a long way.

I remember using it five years ago and getting scared and running back to the comfort of VirtualBox, but now I use it all the time. What are some new features that we can look forward to?

AG: One of the most important data structures in the QCOW2 format is the one that translates guest addresses to host addresses. In other words, to access the disk you first need to read the tables that contain those translations, and, to be performant, that requires a cache. Otherwise you'd have to read those tables twice for every disk access.

Currently, that cache works, but I'm working on some code to make it slightly more efficient. It enables you to have a more fine-grained cache entry, and so that with the same amount of cache memory you can achieve a better level of performance.

LXF: I've heard something about a QEMU advent calendar last year. What was the story there?

AG: A few people in the *QEMU* community created a website that offered a different disk image every day for the month of December 2016. There were some classics in there. I contributed a Spectrum image with seven games developed in this very decade. There was also a tiny chess game, BootChess, so-called because it's small enough to fit into a 512-byte boot sector. There were also some more useful stuff, such as the *WireGuard* VPN and the Rust-based Redox OS. You can, and should, check it out at www.qemu-advent-calendar.org/2016. There's also a 2014 edition, too!

LXF: You can find out more about Igalia and their work with WebKit and Wayland in next month's interview with Alberto's fellow Igalite Juan José Sánchez Penas. **LXF**





The future of package management

Mayank Sharma didn't realise that Linux package management needed a fix... until he ran across projects that are working to reimagine app distribution on Linux.

Package management is often hailed as the single biggest advancement that Linux has brought to the world of operating systems. But under the hood there's a murky and convoluted world of official and unofficial repositories, different packaging formats, lots of metadata, and an endless stream of libraries.

A package manager relies on the format of the packages it manages. So for example, DNF (and previously Yum) uses rpm as the backend and extends the functionality of the backend by adding relevant features. The package management systems save time by automating time-consuming tasks such as resolving dependencies. In fact,

Fedora's DNF started out to address several long-standing issues in Yum including dependency resolution and how it handled online repositories.

Repositories introduce another level of confusion. In addition to official ones, most

“Mixing repositories can have unexpected consequences on the stability of a system”

distros also permit the use of custom and third-party repositories to let packagers push newer releases of software than those available in the official repositories. Mixing repositories can have unexpected consequences on the stability of a system.

Sure, you can use tools like *Alien* to convert between Linux Standard Base (LSB) compliant .rpms and .debs packages. But thanks to the construction of the current packaging systems, you can't really use Fedora without RPMs or Ubuntu without

DEBs. It's safe to say that while the current packaging scheme works, it does leave many avenues for improvement.

Over the years there have been many efforts to retool the Linux software distribution

model to work out the kinks with the existing model. Currently the three that have caught everyone's fancy are AppImage, Flatpak and Snaps. Let's put them under the *Linux Format* spotlight to see if they have cracked the code to app delivery on Linux.

For years, Linux users have lived with the duality of DEB packages for Debian- or Ubuntu-based Linux distributions and RPM for Fedora- or SUSE-based Linux distributions. While these packages provide a convenient way of installing software in their respective distributions, the whole system isn't the most convenient for the application developer. Thanks to the fragmentation aspect, the developers have to create separate packages for multiple distros.

This is something that Appliance attempted to solve back in 2004 (when it was known as klik). Appliance distributes an entire application as a single executable file. The application runs off this package, and doesn't place files on the base system. In addition, the Appliance file contains all libraries and files it needs to run, and this enables them to work on a large number of distributions.

A typical Linux software will create files at various places, requiring root permission to make these changes to the system. Appliance doesn't do this. Technically, an Appliance is a ELF file that also contains an embedded

squashfs filesystem. All the files that are needed to run the application are stored here. When you run the file, the program embedded in this file mounts the filesystem in a directory under `/tmp`. It then starts up the application inside this directory.

For security reasons, mounting a file system still needs root permissions. Fortunately, most desktop distributions have FUSE support, which makes it possible to mount file systems without root permissions. This is used by Appliance files to work seamlessly, and this is the only bit of support that they expect from the base system. Appliances are also very portable and can be run anywhere, including on Live environments. Appliance usually installs a desktop file to integrate it with your Linux installation just like a regular installed application, which means that your software will be searchable through your desktop environment.

Costly convenience

Prolific distribution contributor Neal Gomba has experience with all three formats, because he maintains Flatpak in Mageia, Snaps in

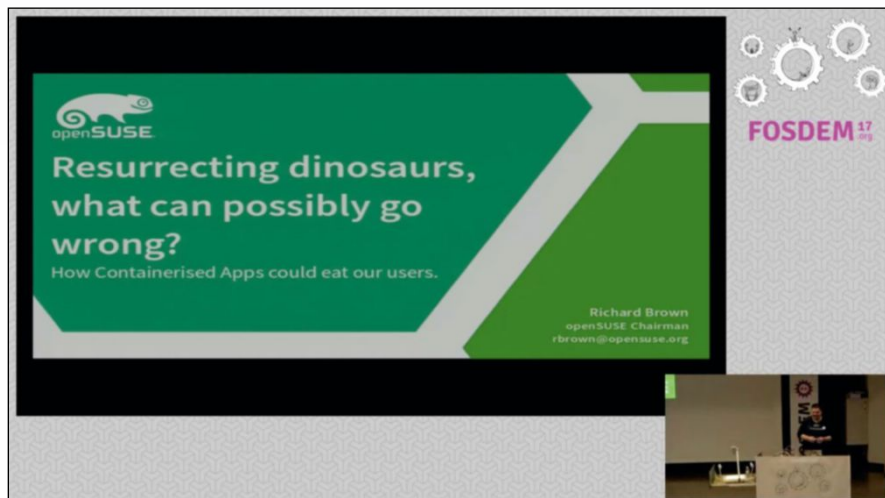
Fedora and has also assembled Appliances. He believes Appliance has no corporate backing because it's "fundamentally flawed" and wouldn't take off unless all Linux distributions agree to provide a base ABI (application binary interface).

"Appliance relies on host libraries to function," he says. "Ironically, this is actually an ideal case, because it maximises the value of what's already on your system and in theory makes it easier to do system integration."

In practice though, bundling libraries leads to incompatibilities that are tricky to debug: "Not to mention, the Appliance founder holds very little regard for security mechanisms and doesn't have a built-in mechanism for Appliance to be confined. The Appliance toolkit (Applimage, ApplimageKit, etc.) provide various extended functionality at runtime and build-time, respectively, but neither of them make it easy or obvious for validating Appliances before using them," as Neal explains.

He's also not sold on the idea of Appliances not having central repositories: "AppStream is functionally useless because Appliances are intended to be downloaded from Joe Rando's Apps o' Fun websites that don't necessarily have to care about integrity. The Appliance founder wants MacOS style "download to run" applications. However, even Apple is moving away from that model now, because it makes it very hard to ensure the integrity of applications and the security of the operating system."

Flatpak, Neal believes, is a marked improvement over Appliance since integrity and security are baked into the design of Flatpak: "It leverages some Linux kernel features to set up basic confinement (seccomp, for example)." The one big difference between Appliance and Flatpak is that the latter doesn't use the host distribution libraries for the applications, and instead relies on runtimes. Neal explains that the runtime is a mini-distribution tailored for



» **Reviewing licenses of the bundled dependencies to ensure compatibility is a key issue with universal formats that doesn't get much attention, says OpenSUSE's Richard Brown.**

Containerised solutions

In addition to Appliance, it's the other distribution agnostic universal packaging formats – Snaps and Flatpak – that hope to make things easier for users and developers.

The one main difference between Flatpak and Snap on the one hand, and Appliance on the other, is that the former run in sandboxed containers. Snaps use a modified AppArmor to isolate the apps, while Flatpak uses SELinux. Like Appliance, Snaps and Flatpak include all libraries and dependencies in the package itself. However, while Snaps can only use libraries included in its package, Flatpak can

use libraries included in the package as well as shared libraries from another Flatpak.

There's a lot of bickering from backers of both the camps extolling the technical virtues of their packaging format over the other. One point repeatedly raised by the Flatpak camp is that Snap was created in-house by Canonical and the technology is hard-coded to use the Snap package store. It's also argued that while Flatpaks are more focused on delivering software to desktops, Snaps are basically just server technology that Canonical has adopted for the desktop.

A comparison of their technological virtues is almost meaningless since the furious pace of development of all these formats is rewriting their disadvantage over the competition. For example, you can now run Appliance apps inside containers created with *Firejail*. The process requires virtually no extra effort from the user, so long as they have *Firejail* installed. Recent versions of *Firejail* even guard against X display server attacks, like key-logging. In fact, proponents of Appliance argue that used together with *Firejail* Appliance apps are even more secure than Snap and Flatpak packages.

» Flatpaks to run on top of. “The default runtimes are based on Yocto,” he says, “but it is possible to build your own runtime on top of a distribution for applications built for that distribution to run on (indeed, Fedora intends to do this).”

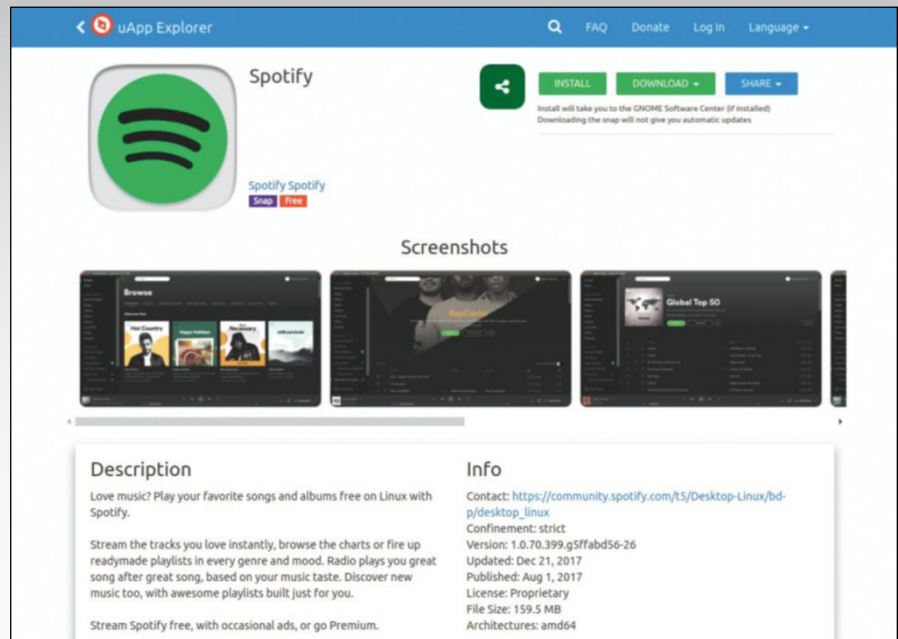
Talking about the security model, Neal says that like AppImage, Flatpak also works for unprivileged users: “They can fetch and run applications from Flatpak remotes like FlatHub freely. The remotes require GPG signatures and SHA512 checksums, so untrusted and non-verifiable Flatpaks are difficult to deliver via remotes. That said, users can just download Flatpak bundles and install them to run too, just like AppImage, but this is not the intended default user model. And even then, Flatpak bundles are expected to be signed so that Flatpak can verify them.”

Desktops domain

Despite all their advantages, Flatpak is only intended for desktop GUI applications. “The creators of Flatpak believe that other technologies (Docker, Kubernetes and so on.) should be used for server applications. And there’s a gaping hole for CLI applications”, Neal points out.

In order to be able to address server applications, Neal says, Snap takes what Flatpak does up a notch and leverages AppArmor for confinement. On the downside however, Snap requires significant work on the distribution that’s integrating it to make it work. “Changes to the kernel are required,” he says, “and there’s an entire class of Snaps called classically confined Snaps that require the distribution to install a `/snap` path that’s either a directory or a symlink to the actual directory configured for the snap mount path.”

Neal points out that to date, the only distributions he knows of that have fully accepted this non-standard path are Ubuntu



» The Spotify music service is an example of Flatpak being used in the real world.

and Solus. Other distributions like Fedora and Arch don’t ship with this path.

All said and done, however, while building Flatpak is a confusing affair, as a suite of tools it’s clear that Snappy is the easiest to use: “However, today, you can only make Ubuntu-based snaps with the official tooling, although there’s work underway to enable other distribution bases,” reveals Neal.

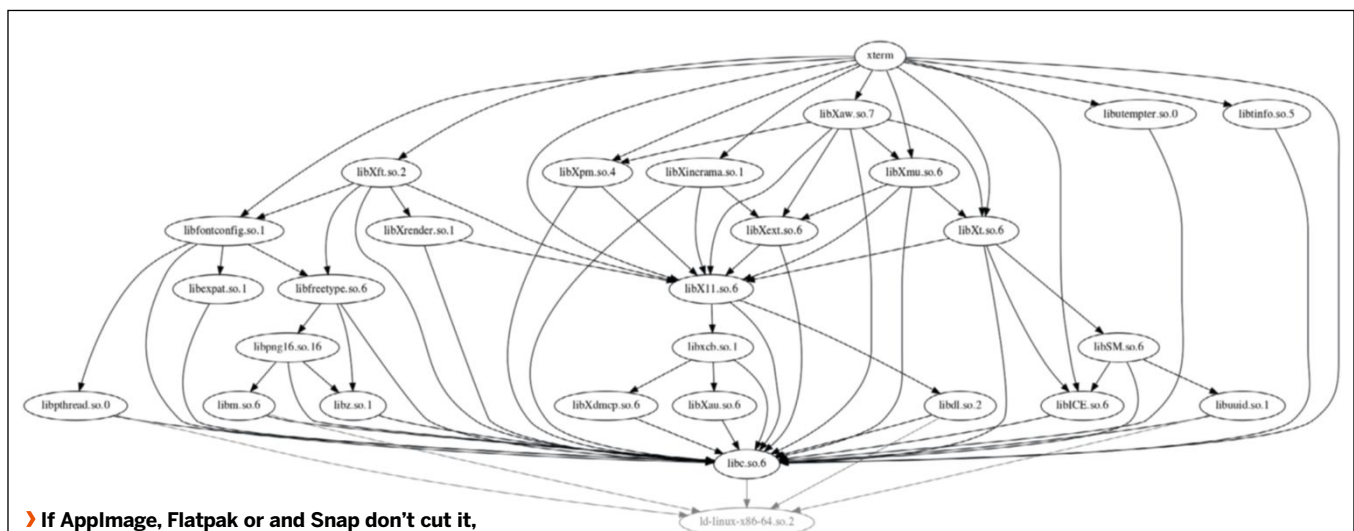
Furthermore, unlike Flatpak, Snappy doesn’t support multiple repositories (or “stores” in snappy parlance). There’s only one Snap store, and it’s run by Canonical. The code doesn’t support easily switching stores, and configuring multiple stores isn’t possible. This means that any user of Snap is effectively tied to Canonical’s store. “This more or less breaks the decentralised model that Free and Open Source software has thrived on in favour

of an Apple-like centrally controlled software delivery platform,” says Neal.

Embracing change

Technical comparisons aside, adoption is the true measure of a technology’s success, which is directly linked to the benefits it provides to users and developers. Jeff Hoogland, lead developer of Bodhi Linux believes that the universal package formats are a reasonable idea, at least for end user-facing applications.

“We already see this same type of idea employed in the gaming world via applications like Steam,” says Jeff, “and so making it easier for non-game applications to install or update without depending on the packaging requirements of specific distros is a big step towards getting more mainstream software onto Linux systems.”



» If AppImage, Flatpak or Snap don’t cut it, follow Genymobile’s example and roll your own universal installer (<http://bit.ly/diy-packet>).

Jonathan Thomas, developer of OpenShot concurs: "I feel very strongly for the need of a universal installer on Linux. Most app developers would love to have consistent branding, messaging, and a single flow for installing and updating their apps across all versions of Linux." The ease of ensuring consistency in installation and updating is one of the main reasons for adopting these universal installers at *digikam*: "Each day we provide a build of all bundles for end user to check if bugs have been fixed between stable release. It's fast and very powerful", says *digikam*'s lead developer Gilles Caulier.

To show the difference between distributing RPM/DEB packages and the universal installers, Gilles notes the project earlier used to receive lots of reports on broken binary compatibility. That's changed with universal installers: "I know how the application runs and I know what are the right dependencies. There's no risk of broken binary compatibility with a bundle – all is already inside."

In fact, when a *digikam* user reports a bug with the distribution RPM, the project asks them to check if the problem is reproducible with a universal installer: "And surprise, 50 per cent of the reports are upstream bugs," says Gilles. "You can imagine the time lost to identify these kind of problems without having a standalone bundle as AppImage to quickly identify a packaging problem."

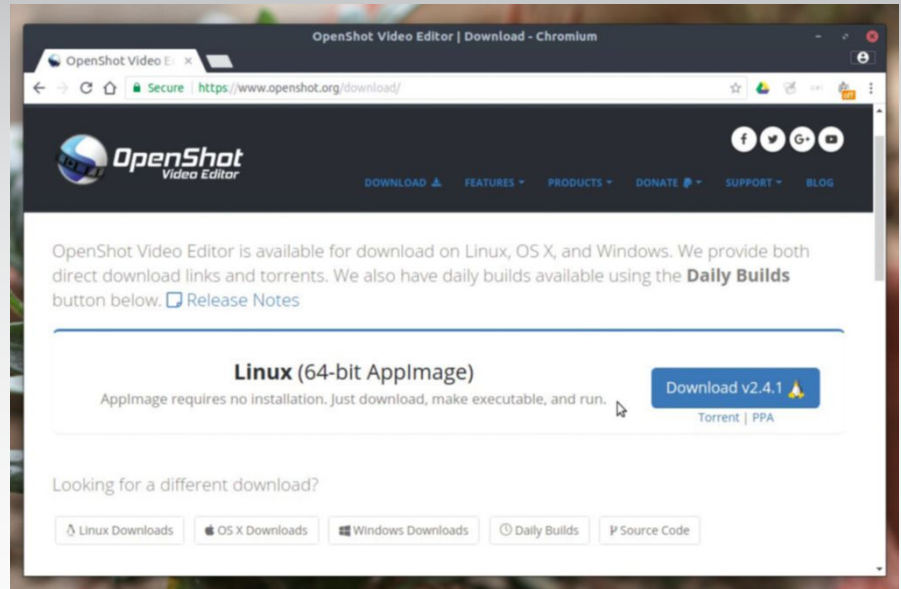
"I chose AppImage because of its wide comparability with different distros"

Jonathan also uses AppImage for OpenShot. "I chose AppImage primarily because of its wide compatibility with different distros, and ease of integration," he says. "I was also familiar with *Krita* (which was also using this format), and was even

contacted by the creator of AppImage who offered to help." Gilles has a similar experience with AppImage. He did fiddle around with Flatpak about a year back, but wasn't impressed with its documentation.

On the other hand, the Solus distribution has employed Flatpak to solve an interesting app delivery issue. In a blog post, lead developer of the project Ikey Doherty wrote that the project, like most Linux distros, distributes the bulk of its software through binary repositories. However, they can't ship some applications, like *Google Chrome*, mostly due to their licensing restrictions. To distribute these apps using Solus' native package management, the apps were first fetched from the vendor and then turned into

» If anyone's looking to throw money his way, Jonathan Thomas of OpenShot would love to create a web-based solution that will do everything from pulling sources, adding dependencies, building a universal installer and publishing it.



Solus' native **.eopkg** package format. The system wasn't robust and provided no mechanism for automatic upgrades. After careful technical deliberation, Ikey went with Flatpak, which he found to be the easiest to integrate into Solus.

No middlemen

Similar reasons were outlined by one of the core KDE developers, Sebastian Kugler, in a blog post earlier last year. Making a case for the three universal packaging formats, Sebastian wrote that currently the project depends on Linux distributors to ship their apps and updates. He argued that this is "problematic" for both distros and application developers because of the delay it causes in pushing updates. Sebastian believes that using the bundled apps/universal packaging formats would solve the problem: "That means we could go as far as shipping apps ourselves and cutting out the distros as middlemen."

Mageia's Donald Stuart disagrees with KDE's position: "Having KDE provide updates so that users can quickly get updates from them could be something worth investigating. However, the QA and packaging teams at Mageia do a very good job of firstly providing updates in a very timely manner; and secondly, ensuring that the updates that are provided are both release stable and work as intended with the rest of the packages included in the distribution."

Neal finds it odd that KDE considers distributions as middlemen given that many KDE contributions come from the various distributions. He believes the basis for KDE's argument is the stability-first development model of distributions. Neal says that unlike Mageia, which follows "the LTS only and

Feature	AppImage	Flatpak	Snap
GUI apps	Yes	Yes	Yes
Server processes	Yes	No	Yes
Run without installation	Yes	No	No
Run without root	Yes	Yes (once installed)	Yes (once installed)
Automatic Updates	No	Yes	Yes
Sandbox	Via Firejail	Built-in	Built-in
Online repository	AppImageHub	FlatHub	Snap store
Individual repos	No	Yes	No
Licence	MIT	LGPL	GPL

» update within the LTS series”, distributions like Fedora and OpenSUSE track upstream releases closely and thus are more actively part of KDE development.

“In my opinion, a large amount of KDE’s frustration comes from distributions that don’t regularly keep their stack up to date,” says Neal. “The Debian family, including many derivatives (excluding KDE neon) prefer ‘stability’, which implies they don’t pull in the

pruning the packages to ensure consistency, compatibility and security. Because this process takes time, many people up and down the Linux app food-chain perceived this as a problem and hailed the universal packaging formats as the solution. We hope we’ve established that while they are useful they aren’t the panacea many imagine them to be.

Mageia’s David Hodgins says that the distro isn’t planning on providing universal packaging formats, though the tools needed to produce them are included for anyone who wishes to produce their own. “They’re a solution looking for a problem to solve, that also introduce additional problems with the testing and distribution of updates,” he says.

“If I could have Snaps with the flexibility of Flatpaks, I’d take such a solution,” says Neal. “But today, while I like Snaps as a system more, I don’t think it’s appealing for a lot of

“Many people up and down the Linux app food-chain perceived this as a problem”

latest releases that introduce features and fix bugs, which frustrates users of the KDE software on those distributions.”

No size fits all

As long-term Linux users know, there’s more to assembling a distro than bundling apps into repositories. It takes considerable effort in

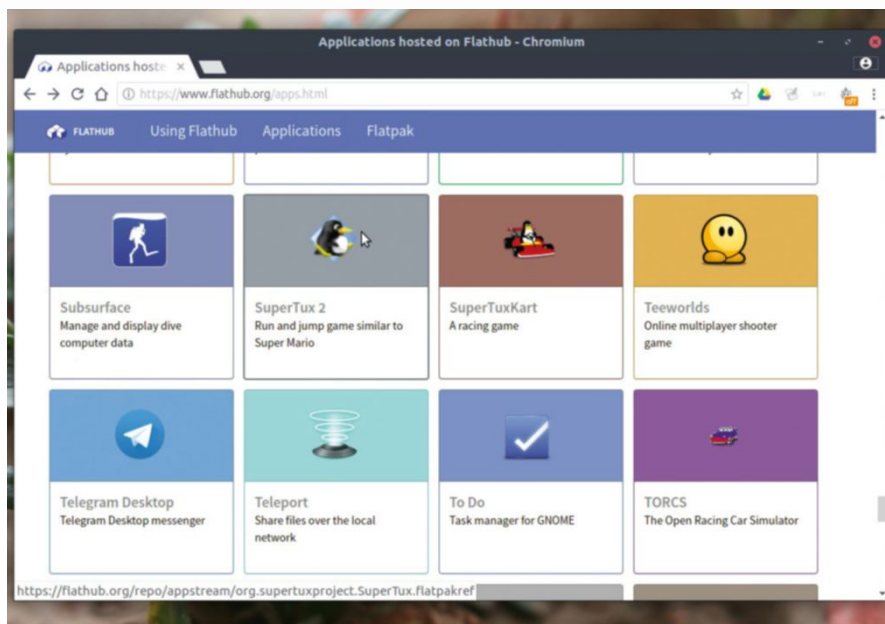


people. Instead, I think Flatpak will win out because it’s easy for distributions to integrate and it solves the app delivery problem for graphical applications, which is what most proprietary app vendors make. And unlike Snaps, since everyone can host their own Flatpak remotes, it’s possible to integrate their own method of user subscriptions or payments to grant access to apps.”

On the other hand, Jonathan would love to see the tools improve for these universal installers: “I still dream that one day, app developers will be able to package applications for all platforms (Linux, Mac, and Windows) with a single universal image, and a single repository/app store, and with minimal effort, and provide a simple, user-focused install/update experience. That is my dream.”

Until that future is realised, the different formats will continue to coexist just like the binary package formats have coexisted. Their pace of development and adoption will continue and a number of mainstream programs will soon be available in one of these formats. From where we stand it appears Snaps will continue its strides with the cloud and IoT apps while Apptainer and Flatpak will divide the desktop apps between them.

“At the end of the day FOSS is all about user choice,” concludes Jeff, “so these types of installers are just another option for end users to give them the flexibility to use their computer as they see fit.” **LXF**



» Flatpak.org is the official build service and app store for Flatpak’ed desktop programs.

Rolling solutions

Part of the reason for new packaging system stems from the link between the current repository-based package managers and fixed distro releases. In a comment on his post, core KDE developer Sebastian Kügler argues that distribution developers should spend their time working on “the things that make a difference.”

Responding to his comment, another long-time KDE and OpenSUSE contributor Luca Beltrame stressed that some of the things which distributions already spend a lot of time on that make a difference include integration, which he argued is something that “neither

Snaps, nor Flatpaks, nor Apptainers will ever do properly.” He also highlighted several other aspects that distributions spent time on while packaging apps such as security reviews, legal reviews and Quality Assurance.

Luca went on to say that the speed of updates of the apps inside a typical distro isn’t constant. In his opinion, the solution to do timely delivery of updates lies in rolling releases. These thoughts were also echoed in a FOSDEM ’17 talk by OpenSUSE’s chairman Richard Brown who argued that rolling releases are perhaps the better model for fixing app

delivery. Using rolling distros, Richard says, enables them to “reuse the knowledge we already have, reuse the tools we already have, reuse the infrastructure we already have.”

He goes on to add that the universal package formats come from the belief that traditional distributions are too slow, which isn’t the case with rolling releases such as OpenSUSE Tumbleweed. The distro is tested thoroughly and still manages to deliver upstream updates in real time, like KDE Plasma on the same day of release, Gnome in under 48 hours, and so on, Richard points out.

ON SALE NOW!

The image shows a large print magazine cover for T3 and a tablet displaying the same digital version. The magazine cover features a red header with the text 'SMART LIGHTING WHY YOU'LL LOVE IKEA'S CLEVER BULBS'. The main title 'T3' is in large white letters on a red background, with 'SMARTER LIVING' underneath. The main feature is 'ULTIMATE 4K LAPTOPS' with a sub-headline 'Cutting-edge notebooks for work or play'. Below this is an 'EXCLUSIVE!' article about 'APPLE'S PHIL SCHILLER' and his talks on HomePod, iMac Pro, and the future. The central theme is the 'TRAVEL GADGET GUIDE', described as 'Essential tech and amazing accessories for your best trip ever!'. This guide is surrounded by various travel-related tech items: a drone, a smartwatch, a camera, sunglasses, headphones, binoculars, a tablet, a train model, and a camera lens. A 'WIN!' badge offers a 'A stereo sound system worth £1,500'. The bottom left corner includes a barcode and the text 'ISSUE 278 PRINTED IN THE UK £4.99'.

SMART LIGHTING WHY YOU'LL LOVE IKEA'S CLEVER BULBS

T3
SMARTER LIVING

ULTIMATE 4K LAPTOPS
Cutting-edge notebooks for work or play

EXCLUSIVE! APPLE'S PHIL SCHILLER
...talks HomePod, iMac Pro, and the future

TRAVEL GADGET GUIDE
Essential tech and amazing accessories for your best trip ever!

WIN!
A stereo sound system worth £1,500

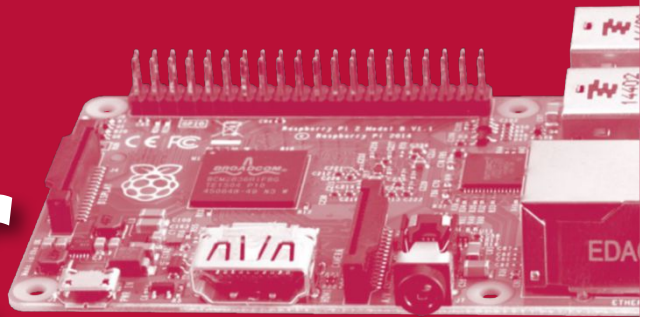
ISSUE 278 PRINTED IN THE UK £4.99

Available at WHSmith, myfavouritemagazines.co.uk
or simply search for 'T3' in your device's App Store

SUBSCRIBE TODAY AND SAVE!
www.myfavouritemagazines.co.uk/T3



LINUX FORMAT Pi user



Giving you your fill of delicious Raspberry Pi news, reviews and tutorials.

LES POUNDER
lives, eats and
breathes exciting
maker projects.



Welcome...

Last night I had the pleasure of attending a lecture hosted by Pete Lomas, co-creator and trustee of the Raspberry Pi Foundation. We learnt about how the Raspberry Pi came to be, and the conversation Pete had with fellow founding member Alan Mycroft while walking through Hyde Park one lunchtime, which focused on the lack of education for Cambridge University's Computer Science applicants. These students had been taught the theory, but lacked the practical experience of writing code to requirements and debugging software problems.

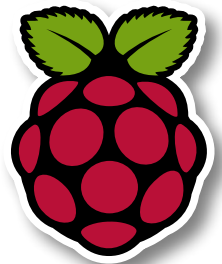
Pete also explained the founders' investments were used to pay for the initial batch of Raspberry Pi, and how they were tested, a few days before Christmas in 2011, only to find a hardware bug which was solved by 1mm of solder to connect the CPU to the power. The Raspberry Pi was almost a non-starter, on the brink of being condemned to the unfinished projects bin of a certain Eben Upton.

But once the Raspberry Pi was ready for development, the task of putting Linux on the Pi fell to the development team of the time, an early Christmas present delivered for them to tinker with.

So as you can see there were many times when the Raspberry Pi may have simply disappeared and become another failed tech startup, but those behind the scenes kept working and trying new ideas.

So what is the motto of this story? Chance meetings, discussing problems and sharing ideas can lead to greater things. Be open with your thoughts as well as your code.

BitScope 3000

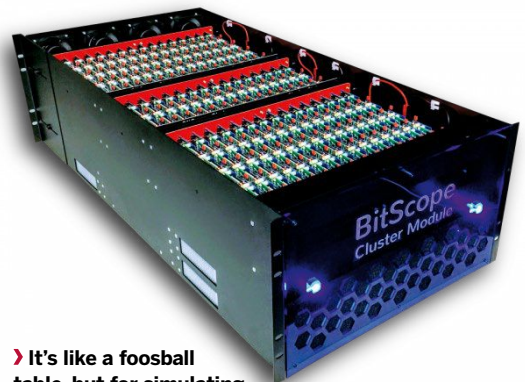


Behold, a test bed for future HPC clusters!

We've seen Pi clusters before, but not like this. Working with the Los Alamos National Laboratory (home of Trinity, one of the world's largest supercomputers), BitScope has crammed 750 Pi 3s into a 35U rackmount case. While it's not going to break any HPC records, the 4kW Pi cluster will inform and inspire the next generation of supercomputers.

Cluster simulations can go a long way to seeing how machines yet to be built will behave, but sometimes this isn't enough, and actual – albeit underpowered – hardware is needed. Rather than risk millions of dollars prototyping these things, the team can learn a great deal from the humble Pi cluster, and use that information to tweak their exascale designs. BitScope (which also makes the Pi friendly micro oscilloscope we reviewed in **LXF194**) also offer a more modest 6U, 144-node cluster for any budding HPC enthusiasts. Learn

more at www.raspberrypi.org/magpi/bitscope-3000-core-raspberry-pi-cluster-computer.



It's like a foosball table, but for simulating machines so powerful they've yet to be made.

Pi Zero WH

Packs pins, just like a – ahem – petite porcu-Pi-ne!

Not content with adding wireless to the diminutive Pi Zero, Team Raspberry has further augmented it with GPIO header pins. Before the WH, users could, of course, solder such a header manually, or solder straight to the board, but now solder-shy hackers can get crafty with LEDs, sensors and other tiny projects. It will also appeal to makers using GPIO Expander on their desktops and laptops. Browse to www.raspberrypi.org/blog/zero-wh for details.

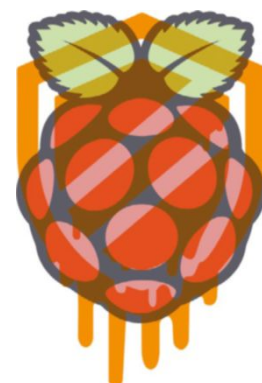


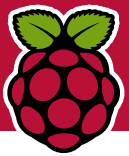
Pi immune

Spectre and Meltdown can't touch this for toffee.

While the whole world panics as the extent of the Spectre and Meltdown flaws becomes realised, we can at least take some small comfort in knowing that all Raspberry Pis are immune from damage. Eben

Upton's blog post lays it all out, and also provides a great high-level view of how the vulnerabilities come about. You can read Eben's post at www.raspberrypi.org/blog/why-raspberry-pi-isnt-vulnerable-to-spectre-or-meltdown.





Vivaldi

Les Pounder loves the science of statistics and now he's found the ideal browser to monitor how many times he's searched for pies and Pis...

In brief...

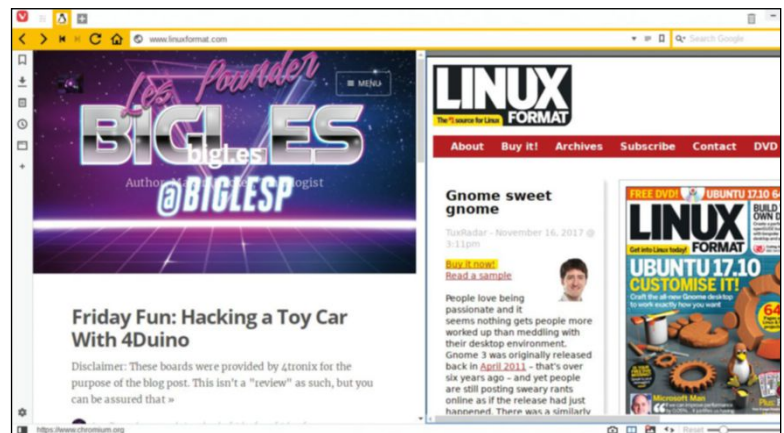
» Vivaldi is a browser that's based on *Chromium*, but it offers a faster and better user interface, and has plenty of customisation options. Vivaldi has a rich set of analysis tools, and users can install *Chrome* or *Chromium* extensions, which provides access to many popular add-ons.

What's this, a web browser for the Raspberry Pi? Does it need another one? Well to be frank, yes. For the first few years of the Raspbian desktop, we used the *Epiphany* browser. While this is a capable browser and well suited to the meagre resources of the Pi, it just wasn't much fun.

Over the past two years we've seen *Chromium* appear as the new default browser. It does a good job of balancing the needs of the user with the available resources, and controversially it saw the inclusion of Flash on the Pi. But when Vivaldi was recently announced with claims that it was a better browser, we had to take a look.

First of all, this is an experimental browser based upon *Chromium* that's been optimised so the user can adapt the browser to their needs. The most obvious example of this is page tiling, where multiple tabs can be kept open in the same window. This is handy for checking multiple sources of information on one screen. Tabs can also be pinned and grouped to keep the user interface clear and easy to use.

Indeed, we were pleased by what the browser's interface had to offer. Its default configuration has the standard address bar and navigation icons at the top of the screen, with a search box (the default option is Bing but this can be easily changed) ready to search for queries. To the left of the window are quick links to Bookmarks, Downloads



» The Vivaldi browser is lovely to use, it provides a consistent user interface, but that interface can also be adapted by the user to meet their individual needs.

and a neat option to make notes on web pages: this can also be done by highlighting any text on a page and clicking the right mouse button. It records the page and time of the note – very handy for educational reference.

Stats the way to do it

The History icon opens the browsing history and also shows how many times a page has been viewed. By clicking the Vivaldi icon in the top left and selecting Tools>History we can see a more detailed analysis of browsing history, which includes a breakdown of the top domains that have been visited, how domains are found (type in the URL, click a link and so on) and a graph showing the times at which we browsed, giving us a clear picture of our browsing habits. Fans of statistics (*hello!-Ed*) are going to love that!

At the bottom of the window is an option to take a screenshot. When we tested this it took rather a long time to render the image. To our surprise we found that Vivaldi had rendered the entire web page into one rather tall image. This feature is interesting as screenshots can be viewed offline later.

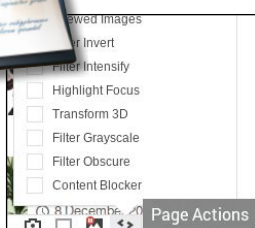
Another feature available at the bottom of the window is the Page Options. This enables you to alter the web page, block content, and render the page in black and white or with a sepia tone. The options also extend to 3D

transforms and skewed images, but we're unsure how this feature can be applied effectively.

But what about the more Raspberry Pi-centric features? Well, even the mighty Pi 3 struggles with web content, particularly animated GIFs, so in the settings these can be turned off, while a "reader" mode enables you to focus on text content. We tried the browser with YouTube and while it played content, it was far from smooth.

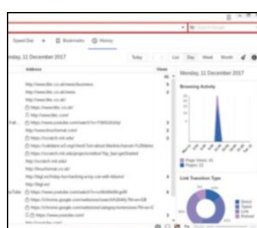
Vivaldi is a great alternative to *Chromium*. It enables users to adapt the browser to their needs and it has great tools to extend the use of the browser. This may be an option of Pi 3, but for those on Pi 2 or older, stick with what you already use for now. **LXF**

Features at a glance



Customise pages

By clicking the <> icon you can change the look and feel of a web page without coding in JavaScript.



Historical browsing

Vivaldi has got all the browsing stats you'll ever need! Find that web page from six months ago!

LINUX FORMAT Verdict

Vivaldi

Developer: Vivaldi Technologies

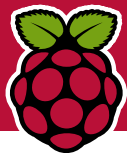
Web: www.vivaldi.com

Licence: CC Attribution 2.5

Features	9/10
Performance	8/10
Ease of use	8/10
Value	8/10

» This provides a powerful user interface, backed by the *Chromium* legacy. Well worth investigating.

Rating 8/10



Raspberry Pi: Amiga Action!

Les Pounder loved his Amiga 500 and in this tutorial he shows us how to recreate the golden years of computing, via the powers of emulation.



Our expert

Les Pounder

works with organisations such as the Raspberry Pi Foundation to promote maker skills. He blogs at www.bigl.es.

You need

- » A Raspberry Pi 3
- » A blank microSD card (2GB or more)
- » Keyboard, mouse and screen
- » Kickstart ROMs
- » Amiga disk images

Back in the 1980s there were a number of home computers from various companies, including Amstrad, BBC and Sinclair. But one company, Commodore, released a computer that transcended the 8-bit era and led to a computing revolution.

The Amiga range of home computers offered something different. An internal 3.5-inch floppy drive, 512kb of RAM that was expandable using an add-on board, and compatibility with peripherals to manipulate and create content for television (the early series of *Babylon 5* used content generated on an Amiga 4000 with the Video Toaster add on).

The Amiga also had a healthy magazine following, including *Amiga Format* whose erstwhile editor Nick Veitch went on to found a periodical called *Linux Format* (*Amiga Computing 4ever!*—Ed). In this tutorial we shall have a little fun and create our own Amiga 500 using a Raspberry Pi 3, and then play a classic game!

Fat Agnus!

For this project we'll be using Amibian, a Raspbian-based operating system for Amiga emulation. It includes a user interface for the UAE (Useless/Universal/Unix Amiga Emulator... it has many names!) that enables us to configure the emulator. To download Amibian head to <https://gunkrist79.wixsite.com/amibian> and click Download and extract the image file from the archive.

As well as downloading the emulation software, we also need to download *Etcher*, which is an easy-to-use SD card imaging tool. It's a graphical tool for those that find `dd` a little

daunting. Download *Etcher* from <https://etcher.io> and then extract it.

To use *Etcher*, go to the folder where it's been extracted and double click its icon. *Etcher* is a self-executing file, so it can be directly used. Once *Etcher* opens you have three steps to follow. The first is to select the image to write to the microSD card. In this case it's the Amibian image. Next, select the drive to which the image should be written. Typically, *Etcher* will correctly identify the drive to use because it looks for large hard drives and ignores them in favour of smaller SD cards. Finally, clicking Flash will write the image to the microSD card. Note that you'll need to enter your password in order to flash the image!

The flashing process should take no longer than five minutes. When it's finished, close *Etcher* and remove the microSD card, but then pop it back in so that it can be mounted for use.

Our next step is to obtain the Kickstart ROMs (the BIOS of the Amiga) so that we can boot the emulator. If you have an Amiga already then there are ways of obtaining your own image of the ROMs, but the easiest and legal way to get the ROMs is to purchase the Amiga Forever Essentials app for Android (www.amigaforever.com/android) and then copy the ROMs to your computer. The app costs £1.39 and it provides legal access to all of the Kickstart ROM files for the Amiga 500 to the 4000 (OS 1.3 to 3.1).

With the Kickstart ROMs to hand, we now need to copy them into the Kickstart directory of the Amibian SD card. This directory is in the largest partition of the microSD card, and it's in `/root/amiga/kickstarts`. Because this folder is owned by root, we need to either copy the files using the terminal as root, or run the file manager as root. Choosing the latter means we can drag and drop the files to the correct folder:

```
$ sudo -i nautilus
```

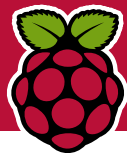
Not again Paula!

The Amiga primarily used floppy disk (3.5-inch, 880kb) and these disks contained games, applications and the operating system called Workbench. If you already have your disks to hand, you can create images of them using software on the Amiga. There are many websites offering ADF files (Amiga Disk Files) that are images of floppy disks, but their legality is dubious at best because they're not strictly "abandonware". So we'd advise caution if you follow this route.

For this tutorial we already had an ADF of Workbench 1.3, the classic strategy game *Cannon Fodder* and an original *Amiga Format* issue 10 cover disk. We used *EasyADF* (<http://bit.ly/easy-adf>) to create ADF images of our



» *Cannon Fodder*, a simplistic real-time strategy game, was a popular Amiga title.



Creating the ultimate Amiga

In 1987 an Amiga 500 would cost around £500, and that was a lot of money in the 80s! The top-of-the-range Amiga 4000, a more powerful machine with better CPU, an Advanced Graphics Array (AGA) chipset for better graphics, and internal IDE hard drive/CD-ROM was released in 1992 and that sold from \$3,699. So to have the most powerful Amiga in 1992 would have seriously stretched your finances!

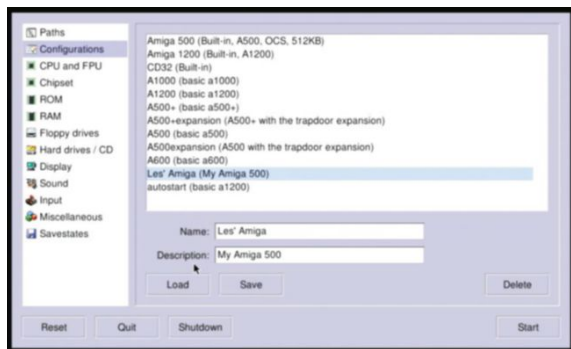
But in 2018 we can recreate a similar spec A4000 for only £35. All we need to do is bump up the RAM to between 2MB and 18MB, set the CPU to a 68040 at 25MHz (or Faster!) and change the Chipset to AGA.

We can also add a hard drive to our setup, and install the Workbench operating system onto the drive ready for use. In the Amibian user interface there's an options called Hard drives/CD and

from here we can configure a device (USB flash drive) or create a hard file (a hard drive volume as a single file) onto which we can then install our operating system. Obviously, you'll need a Kickstart 3.1 ROM and the Workbench 3.1 disks in order to unleash the power of the A4000.

Here's a great video explaining the process of creating a hard drive for Workbench 3.1:

<https://youtu.be/JybUir6ius>.



» **Configuring your Amiga is made easy thanks to the user interface. Here we can increase RAM, CPU speed, create hard drives and swap floppy disks.**

floppies. Then we copied the ADF files to the **floppys** directory located in **/root/amiga/floppys**.

With those copied across, unmount the microSD card, insert it into the Raspberry Pi 3, connect your peripherals and boot into Amibian.

Kickstart me

On first boot we see the *Amibian* logo, and then after only a few seconds the user interface appears. To the left are all of the configuration options available (RAM, ROM, CPU and so on) But for this first boot, let's click Quit to take us to the command line. We need to use the *Amibian* command line, so from the text menu select option 6 (Settings) and press Enter. Then in the next screen type the following to use *raspi-config*:

```
$ raspi
```

In *raspi-config*, select the menu option to expand the filesystem and then press Enter to start. For the change to take effect we'll need to reboot and return to the *Amibian* user interface.

Denise hold and modify

The Amiga 500 came with 512kb of RAM (upgradable to 9MB) and the CPU was a Motorola 68000 running at 7.09MHz. Despite this rather meagre-sounding spec, the Amiga was a powerhouse of its time that was capable of multitasking operations, playing digitised video and displaying up to 4,096 colours on screen (in HAM mode).

We're going to configure a basic Amiga 500 with 1MB of RAM. First, go to Configurations, create a new configuration as <YOUR NAME> Amiga and click Save. Next, move on to CPU and FPU and set the CPU to 68000, FPU None, and CPU Speed as 7MHz. Then move on to Chipset. For this, select

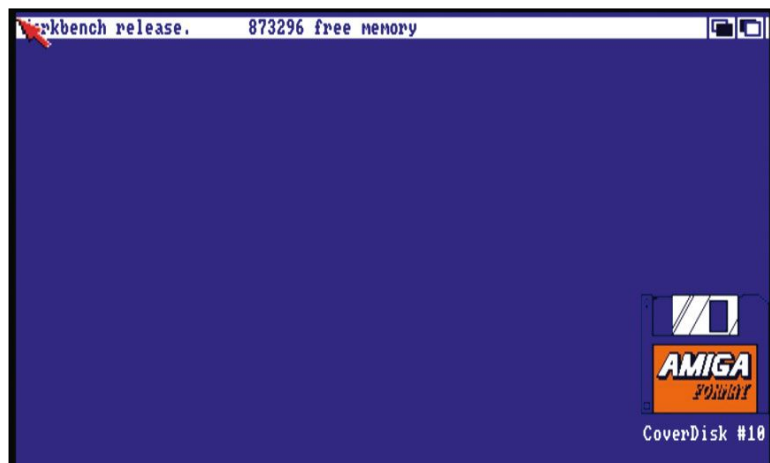
OCS, Blitter to Immediate, and don't tick the the Copper box. Next, move to ROM and using the ... button open the dialog and navigate to **/root/amiga/kickstarts**. You'll see the Kickstart ROMs that we copied there earlier. Select the **amiga-os-130.rom** (or whatever your 1.3 Kickstart ROM is called) and then press OK. Now set the RAM for the Amiga. Set the Chip RAM to 1MB (an expanded A500.)

In order to play a game we need to insert the floppy disk(s) into the Amiga. The Amiga could have up to four floppy drives (one internal, three external), and for our demo we played *Cannon Fodder* which came on three floppy disks. So we enable DF0 (internal) and used the ... menu to locate the game's ADF image that we want to play. The last configuration step before playing the game is to configure a joystick. If you have a USB joypad/stick then Amibian should detect it and configure it, but you can tweak it using the Input configuration option. Remember to save your setup for your next gaming session using the Configurations option.

Rock Lobster time

After all that tinkering we can finally play the game. *Amibian* will emulate the speed of an Amiga floppy drive, so don't worry – your game will load in a minute or so! If you can't wait, click F12 to open the user interface, go to Floppy Drives and change the emulation speed to suit your patience. F12 can also be used to change configurations, load a new floppy disk and reset the Amiga should the game crash.

So there we have it, an emulated slice of video gaming and computing history, and something that brings back many wonderful memories for us at *Linux Format*. **LXF**

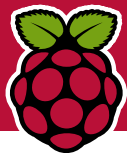


» **Here's the coverdisk from issue 10 of Amiga Format. At the time coverdisks were the most popular way to share information about applications and games!**

Quick tip

The Amiga computer series came in many configurations. The most compatible with games was the Amiga 500 Kickstart 1.3, but for games that require enhanced graphics and power, the additional grunt of the AGA chipset, 2MB RAM and 68020 CPU will be needed!

» **Get your Pi filling here** Subscribe and save at <http://bit.ly/LinuxFormat>



Twitter: Bot fun

Nate Drake sets up a Twitter bot on his Raspberry Pi to spam the net!



Our expert

Nate Drake is a technology journalist specialising in cybersecurity, retro tech and chatting with Russian bots for fun and profit!

The simplest description of a Twitter bot is a program designed to produce automated posts on Twitter. Given how simple it is to tweet yourself, it's worth explaining why people go to the trouble of doing this.

The most common use for bots is for (*Russian?*—Ed) spam. The bot examines key words – for example, “cleaning products” – and then responds with a promotional link for people to click to be taken to the spammer's website.

This feature has non-spammy uses, however. Bots can be programmed to search for any word or phrase, and respond accordingly. One entertaining implementation of this was @BDZNappa, which would search for people tweeting the phrase “over 9,000”, to which it would always respond, “WHAT!? NINE THOUSAND?” to the person in question.

Bots can also tweet from a text source, such as @SunTzuBot, which tweets daily quotes from *The Art of War*. More sophisticated bots, such as @JustDiedBot, actively search the internet for source material. @JustDiedBot searches Wikipedia for information about recent deaths, and tweets RIP announcements as and when they happen.

Bots have practical uses, too. They can be programmed to tweet at regular intervals, so can be used as a ‘dead man's switch’ to tweet a message unless you reset it every day. It's also possible to schedule a tweet for a future date, so you can use the bot to send reminders.

Quick tip

Once the access token has been generated, you can delete your mobile phone number from this account. Visit <https://support.twitter.com/articles/81940> for details..

Pi Bot Ready!

If you find any of these possibilities intriguing, you need to set up a dedicated Twitter account for your bot. Don't be tempted to use your existing Twitter account for this if you have one, because repeated tweets could be mistaken as spam by Twitter, and your account could be suspended.

In order to proceed, we need a new Twitter account with a confirmed mobile phone number. If you already have a Twitter account, visit <https://support.twitter.com/articles/81940> and follow the steps there to delete your mobile phone number from your account for the time being.

Next visit www.twitter.com and choose Sign Up. You can use any email address you wish, provided it's not already

registered with Twitter. Choose a name that's appropriate for the bot. For the purposes of this tutorial, we'll create a bot named Sherlock Bot, with user ID @holmesbot1.

You're asked to provide your phone number and then confirm it by SMS. Twitter requires this for bots to reduce the likelihood of spam.

Once your Twitter account has been created, you may need to click Continue a few times at the top-right to skip invitations to import your contacts or follow other users.

Next, we need to create a Twitter application that will enable your Pi to access your Twitter account. Go to <http://apps.twitter.com>, then click Create New App.

You're asked to fill in the Application Details. For the purposes of this tutorial, we're creating a bot that regularly tweets the Sherlock Holmes stories, but you're welcome to change the Name and Description as you see fit.

Under Website, for now simply put www.twitter.com. Leave the field named Callback URL blank. Tick the box to say you agree to the Twitter Developer Agreement, then click the grey button marked Create your Twitter Application.

Make sure Access Level reads Read and Write. If not, click Modify App Permissions to change it.

Next click Manage Keys and Access Tokens. Scroll to the bottom of the page and click the grey button marked Create My Access Token.

You'll see a message saying the access token has been generated. Keep a copy of this page in a safe place or leave it open in your browser, because we will need it shortly.

Next, open the terminal app on your Pi or connect to it via SSH. First, we need to install some extra software by using the following command:

```
sudo easy_install pip
```

Next, create a directory for the bot and open it:

```
mkdir holmesbot1 && cd holmesbot1
```

For this example, we're going to a specialised Python library created by Edwin Dalmaier, named Markovbot. The software essentially takes some text from a source (in this case *The Adventures of Sherlock Holmes*) and randomly constructs plausible looking sentences with it.

You need to download and unzip the software with the following command:

```
wget https://github.com/esdalmajer/markovbot/archive/master.zip && unzip master.zip
```

Move to the new directory with `cd markovbot-master` and install more required software with these commands:

```
wget https://bootstrap.pypa.io/ez_setup.py && sudo python ez_setup.py
```

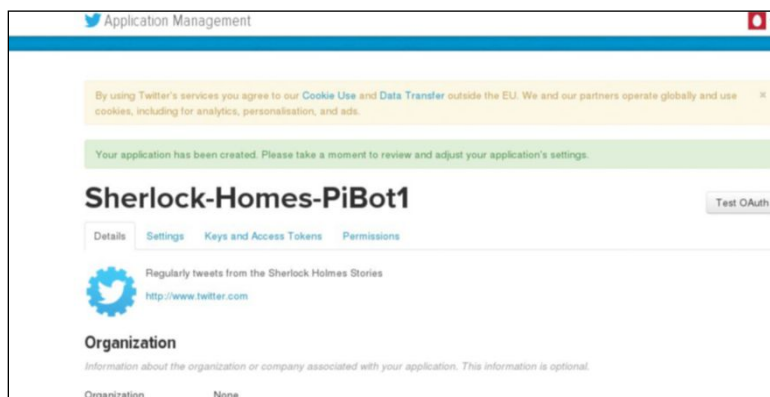
```
sudo easy_install twitter
```

Next, we need to download a text file to use as the source for our random tweets. This file comes from the Project Gutenberg website, but feel free to use any TXT file you like:

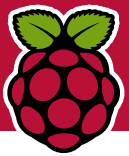
```
wget http://www.gutenberg.org/cache/epub/1661/pg1661.txt
```

Then we create an empty file to place our code. You can choose any name you like, provided that you use the extension `.py` at the end:

```
nano sherlock1.py
```



➤ Once the application has been created, scroll down to enable the access tokens. This enables the bot to log in to your Twitter account.



Bots of note

Bots have been around for several years, and some even have thousands of followers. Some of them simply react to words in tweets. For instance, anyone using the phrase “illegal immigrant” can expect a response from @DropthelBot with the message, “People aren’t illegal. Try saying ‘undocumented immigrant’ or ‘unauthorised immigrant’ instead.”

The Twitter bot @everyword began tweeting every word in the English language in 2007. A new word was tweeted every 30 minutes until it completed its task in 2014, after 109,157 words. The author even published a book of the event.

There are also bots that exist for political parody. Mentioning the words “communism” or “socialism” provokes the ire of @RedScareBot

Robot J McCarthy himself, who tells you the “Red Storm is rising” and to “circle the wagons”.

@DeepDrumpf is a Twitter bot created by MIT, which uses neural network technology to analyse data and post tweets in the supposed speaking style of Donald Trump. The developers claim the bot was trained using transcripts of Donald Trump’s speeches.

The twitter bot @factbot1 makes good use of images. Creator Eric Drass programmed the bot in response to the tendency of some people to believe unproven facts, provided they’re accompanied by an image. The bot regularly tweets nearly true and nonsensical facts. One such ‘fact’ is that the Canadian government derives 38 per cent of its income from the sale



➤ **Robot J McCarthy is here to remind us of the supposed perils of the Hammer and Sickle.**

of doughnuts. This is plausible, given the ubiquity of Tim Hortons cafés in the land of the maple leaf, but sadly false.

Enter the following code in the new file:

```
import os
import time
from markovbot import MarkovBot

#####
# INITIALISE

# Initialise a MarkovBot instance
tweetbot = MarkovBot()

# Get the current directory's path
dirname = os.path.dirname(os.path.abspath(__file__))
# Construct the path to the book
book = os.path.join(dirname, u'pg1661.txt')
# Make your bot read the book!
tweetbot.read(book)

#####
# TWITTER

# The MarkovBot uses @sixohsix' Python Twitter Tools,
which is a Python wrapper
# for the Twitter API. Find it on GitHub: https://github.com/
sixohsix/twitter

# ALL YOUR SECRET STUFF!
# Make sure to replace the 's below with your own values, or
try to find
# a more secure way of dealing with your keys and access
tokens. Be warned
# that it is NOT SAFE to put your keys and tokens in a plain-
text script!

# Consumer Key (API Key)
cons_key = 'yourconsumerkeyhere'
# Consumer Secret (API Secret)
cons_secret = 'yourconsumersecrethere'
# Access Token
access_token = 'youraccesstokenhere'
# Access Token Secret
```

```
access_token_secret = 'youraccesstokensecrethere'

# Log in to Twitter
tweetbot.twitter_login(cons_key, cons_secret, access_token,
access_token_secret)
# Start periodically tweeting. This will post a tweet every
minute.
# (You're free to choose your own interval, but please don't
use it to
# spam other people. Nobody likes spammers and trolls.)
# This function operates in a Thread in the background, so
your code will not
# block by calling it.
tweetbot.twitter_tweeting_start(days=0, hours=0, minutes=1,
keywords=None, prefix=None, suffix='#IamSherlocked')
```

Once the code has been entered, you need to use your arrow keys to navigate to ‘yourconsumerkeyhere’, ‘yourconsumersecrethere’, ‘youraccesstokenhere’ and ‘youraccesstokensecrethere’ and replace these with the values from Twitter’s website that you noted down earlier. Note that you need to leave the quotation marks in place when replacing the values.

You’ll also notice the filename `pg661.txt` under the words `Construct the path to the book`, so you also need to change the filename to your own text file.

Scroll to the bottom of the code and note that, by default, this code will tweet every minute. Change this if you wish – for example, to once a day: `days=1, hours=0, minutes=0`. Finally, you may wish to change the suffix placed after every tweet to something else or to `None`.

Press Ctrl+X when done, then Y, then Return to save. You can run your script at any time with the following command:

```
sudo python sherlock1.py
```

Be sure to substitute `sherlock1.py` with the actual name of your file. The terminal will show the tweets, but you can also check it on the website.

There are many more possible Python projects you can do with Twitter, including responding to keywords, automatic retweets and even grabbing values from web pages, such as stock market prices. Head over to www.raspberrypi.org/blog/tag/python to see some of the exciting projects that are done with Python. **LXF**

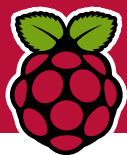
Quick tip

Head over to www.gutenberg.org for free ebooks. There are various book formats. Make sure to choose the link to Plain Text UTF-8 to be sure the bot can read it.

Quick tip

If you’re feeling super lazy you can head over to and grab the code straight off the internet <https://pastebin.com/SnJi18rd> now there’s a quick tip!

» **Love your Pi more** Subscribe and save at <http://bit.ly/LinuxFormat>



Hydroponics: Monitor plants

A case of green fingers is nothing to worry about, as **Kent Elchuk** gives you the goods to monitor a garden with images, video and a moisture sensor.



Our expert

Kent Elchuk is a full-time web developer and Linux enthusiast whose spare time includes programming and hydroponic food production.

This month's tutorial will be a guide to growing healthy, hydroponic (or organic hydroponic) food with the aid of a Raspberry Pi. Hydroponics has many advantages compared to other, more conventional methods of food harvesting, including faster rates of growth and improved water conservation.

Because plants take up their 17 essential nutrients in liquid form, going down the hydroponics route enables gardeners to apply a precise diet to their crop, while using a sterile medium for the root zone that's free of pathogens such as *E. coli* and salmonella.

Now, some of you may be wondering what the heck does a Raspberry Pi have to do with hydroponics or organic hydroponics? Well, it's such a useful device that it'll carry out two tasks: monitor the garden remotely via a webcam, and detect moisture levels.

Although Linux technology and growing vegetables are the focus of this tutorial, we'll also briefly explain the garden setup and feeding procedure. This will enable you get a grasp how it all ties together.

Although there are many webcams that work out of the box, two very common, affordable cams that often come on sale are Logitech's C170 and C270 models. These can be bought via Logitech's website (www.logitech.com) for £17 and £26, respectively.

From the moisture detection side of things, we'll be using a Soil Moisture sensor available from the Mod My Pi website (<http://bit.ly/soil-moisture>). The device costs a very reasonable £4, and can be hooked up in minutes. It connects to the Raspberry Pi GPIO pins and the other end is inserted into the plant pot.

As far as accessing the updated watering status goes, two methods will be covered. One uses our Raspberry Pi as a web server; the second accesses a remote website address.

Grow a web server

Let's assume we don't have a remote web hosting account to transfer files. No problem: we can use our Raspberry Pi and access it through a browser via your IP address. Every home internet plan sets us up with an IP address. Some are fixed and some are static.

We can see our IP address by opening the router software. However, an easier solution is to visit <http://myipaddress.com>, which will give us the information we need.

Once we know our IP address, we can type it into the browser to see the stream. Note that port forwarding is required to see the web content remotely using our IP address. This can be configured with a router login.

As we mentioned at the beginning of the article, the Soil Moisture sensor will be used to keep an eye on the watering side of things.

Let's start with the hookup. To assemble the moisture sensor, we connect it to three GPIO pins on the Pi: GPIO 17, power and ground. Afterwards, we connect the sensor board to the metal plates of the moisture probe. The entire kit comes with all the parts. The photo (*below*) shows how everything needs to be connected.

With the hookup in place, the unit can check for moisture right away. Thus, all we have to do is stick it into the ground and water a plant. If it detects moisture, two lights illuminate. If not, only one light shines. By default, the output is digital and it uses a simple on/off procedure to detect moisture. With a little time and effort we could make this setup work for analog output. But we'll keep things simple for this tutorial.

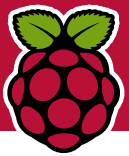
This device can detect when fresh water is added to the plant. But it's not great after that. In an hour or so, after the water has drained away, the moisture detection light goes off.



► A USB cam is attached to a USB port. The moisture sensor connects to GPIO 17, power and ground. The sensor board is connected to the metal plates of the moisture probe.

Quick tip

Home servers aren't an exact science. Some internet service providers enable port forwarding for home-hosted websites, while others block the port and others. Thus, we can always use a cheap web host and transfer the data to a remote server to solve the problem, or change the ISP.



Garden in Motion

First off, let's dive into the setup and details of plant monitoring. To do so, we simply install *Motion* using this command:

```
sudo apt install motion
```

After *Motion* is installed, we need to edit the file `/etc/motion/motion.conf`. After it's opened in our usual favourite editor, we just need to change the odd word, such as swapping 'on' to 'off'.

Although the ordering of the lines is spread throughout the file, the listing below shows what these lines should be. If we use the *Vim* editor, we can use the forward slash and word followed by pressing Enter to find the appropriate line that contains the desired words(s). (or with any other sensible editor *Ctrl+F-E*)

```
daemon on
width 640
height 480
framerate 5
ffmpeg_output_movies on
snapshot_interval 60
stream_localhost off
webcontrol_localhost off
vi /etc/default/motion
start_motion_daemon=yes
```

Asides from that, let's make sure it starts automatically on reboot:

```
update-rc.d motion enable
```

The next step is to add an *Apache* web server and PHP; if none exist on our Pi. PHP will be used if we send files to a remote server and for the simple script used to monitor the watering.

During installation, make sure to type Y and press Enter to ensure the installations occur.

```
sudo apt-get update
sudo apt-get install apache2
sudo apt-get install php7.0 php-pear libapache2-
mod-php7.0 php7.0-mysql
sudo apt-get install php7.0-curl php7.0-json
php7.0-cgi
```

Now, if we reboot the Pi and open the Pi network address like 192.168.1.777:8000, we should see the live stream.

So, if this article was geared towards houseplants, and the light goes out an hour after watering, what good is it if we only need to water them once a week? Not much, to be honest. However, since this article is about hydroponics, the setup is just fine because the plants are on a once- or twice-a-day feeding cycle. Thus, we can easily work with the detection on this level and it'll be as successful as any other device. The water detection can be monitored remotely because we'll be running a daily timer each day to feed the plants with a pump.

The key to this project is that we use a *cron* job to check the output as either on or off. If it finds that it's on then we'll receive an alert. This way, we know whether or not irrigation takes place.

So, if we're on the road and we receive an alert at the expected time, we can be confident that our plants won't be flopping over from wilting. This alert is especially useful for the last feeding of the day so we can have a sound sleep, or, don't need to call someone to take care of the plants.

Moisture alert!

So, let's get down and dirty into the code for detecting moisture and making an alert. Such a small script can be made with Python or PHP.

In our case, we'll use PHP because it has a `shell_exec()` function that enables us to use raw Linux shell commands instead of using a GPIO library and its functions with Python. Note though, that PHP also can include a GPIO library that includes an additional install.

So, let's take a look at the code below and copy it into a file called `moist.php` located in the `/var/www/html` folder. The detailed explanation will follow afterwards. This file can be accessed using the URL `http://myipaddress/moist.php`. Thus, this file serves a dual purpose: to see the monitoring data output and take a sensor reading.

```


<?php
$on_off = shell_exec('gpio read 0');
echo $on_off;
if($on_off == 1){
```

```
echo "<br/>It is off";
$fp = fopen('sensor.txt', 'w');
fwrite($fp, $on_off);
fclose($fp);
}else{
echo "<br/>It is on!<br/>";
$fp = fopen('sensor.txt', 'w');
fwrite($fp, $on_off);
fclose($fp);
}
```

Here's how it works. The first two lines display the live stream and latest image taken from the webcam.

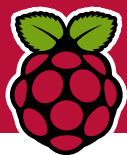
Then, the Linux GPIO library that comes shipped with the Raspberry Pi reads the sensor and returns '1' if dry and '0' if moist.

Just in case we want to record the data, the `fopen()` line up to the `fclose()` line will rewrite a file called `sensor.txt`, which basically stores the 0 or 1 taken from the reading. »



» Aside from a pump, 1/2-inch tubing, 1/2-inch feeder lines, 1/4-inch barbed fittings and various 1/2-inch PVC fittings are used to make the feeding system.

» We ♥ gardening! We're old. Subscribe at <http://bit.ly/LinuxFormat>



» Now that we have the code, let's set up a *cron* job that will runs automatically. In our case, we want it to run twice a day at 10am and 3pm, which is a feeding schedule this author has used in the past. Thus, the five pieces of the puzzle would like

```
0 10,15 * * *
```

The example below shows a *cron* jobs every minute; one that runs at 10am and 3pm. For those new to *cron* jobs, let's explain how it works. There are five times to set from left to right: minute, hour, day of month, month, day of week. After the schedule, the command is written as follows:

```
0 10,15 * * * php /var/www/html/moist.php >> /var/www/html/moist.txt 2>&1
```

Now, a quick note about the images. Because that path to display the image is for the **/var/www/html** folder, we need to copy the file from **/var/lib/motion** and give it proper permissions. This procedure will be performed with a *cron* job as shown below:

```
* * * * * cp /var/lib/motion/lastsnap.jpg /var/www/html/test.jpg && chown pi:pi /var/www/html/lastsnap.jpg
```

Here are a few tips regarding the **moist.php** file. It will check the current status of moisture and write to a file whether you run it in a browser or from a *cron* job. Towards the end of this article, there's another block of code that just reads the **sensor.txt** file and displays the result. Therefore, you could always run the **moist.php** *cron* job from a path outside of the web directory (for example, **/home/pi/moist.php**) and just display the output (moisture reading and image) from the **/var/www/html** directory. This is our recommended option.

Remote servers

The plan is to transfer the data to a remote server, but we still need the previous setup for the Raspberry Pi because it takes the photos and reads the watering data. So, for the gardening

enthusiast who wants to transfer and read the water output and the latest image to a web server, this section will explain that procedure.

File transfer

Running *Motion*, the saved files are timestamped and the most current file is always called **lastsnap.jpg**. This makes it easy to always keep tabs on the situation. If we have *Motion* up and running, we can look inside the **/var/lib/motion** folder and see all the images that had been taken every 100 seconds.

If images are absent in the **/var/lib/motion** folder, we must make sure it has permissions for the *Motion* user. Thus, the command **chown -R motion:motion /var/lib/motion** can fix this issue quickly.

There are various methods for transferring files from the Raspberry Pi to the remote server: SCP, FTP, SFTP or FTPS to name just a few. If we keep things simple and are managing both servers with a single admin, then any of these methods are good to use.

On the other hand, if we have cameras from various foreign networks and want all the images on the same remote server, then this will require more work. To do that, we can create FTP accounts for every Raspberry Pi that is carrying out the monitoring. Then, each Pi will authenticate and upload the file to its own account.

In the case of multiple machines, each unit will need its own username and password, and the file will be transferred with the **curl** command. That's about it, because once the server interprets the user and password, it already knows which folder will receive the file.

Green FTPing

Before we go to much further, let's take a look at the first command, which is a basic FTP transfer. We give the file called **send.sh** executable rights and the rest just takes care of itself.

In addition, always remember that the file must be executable so it can run. Take a look at the command below and the actual code block shown after that. After we see these two snippets of code, I will explain the ftp process, which does the same operation like a tool such as *Filezilla* except automatically.

```
chmod +x /var/lib/motion/send.sh
```

```
#!/usr/bin/env bash
```

```
PASSWORD=Member111
```

Quick tip

Cron jobs have five time intervals that are set at the beginning of each job. From left to right they are: minute(0-59), hour(0-23), day of month(1-31), month(1-12) and day of week(0-7).

» Although plant stakes work great, this 2 litre per hour dripper is supported with a cut-off zip tie which cost only pennies.



The basics of hydroponics

Quad pots can be stacked on top of each other. Pots containing tomatoes can be placed two or three high, while peppers, lettuce and cucumbers can be five or six high.

The pump delivers solution through 0.5-inch poly tubing that has an end cap. Running along the header line there are barbed fittings that are inserted into the header line, which connects to smaller, 0.25-inch tubing. This, in turn, runs to the pots.

At the end of the 0.25-inch tubes are two-litre per hour drippers. The drippers are held with

plant stakes. Two drippers in the top pot and individual lines to the lower pots works well. At the very bottom of the system are single three-gallon pots that can also be planted.

As far as media goes, we can use any hydroponic media of choice, such as a soil-less mix and coco fibre/coir. Any hydroponic plant food will work too, but, the powder option is cheapest. A common food-grade 55 gallon drum can acts as the reservoir.

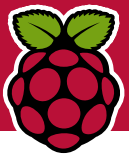
Two tools that are in most hydroponic growers' toolbox are a ph pen and EC/TDS

conductivity meter. However, we can reduce costs and just use ph test strips or a liquid test.

A ph of 6.0-6.5 can meet general vegetable needs, while 1,500-1,800ppm (parts per million) works for most vegetables, although lettuce will be fine at 400-500ppm.

Because plants that are grown hydroponically can produce large yields, staking is required for vegetables such as bell peppers, tomatoes and jalapeños. Lettuce and cucumbers are low maintenance, while many varieties of tomatoes can simply hang down.

» **Get more fun projects** Subscribe and save at <http://bit.ly/LinuxFormat>



► Each column has stacking pots. Each pot holds four plants and columns can contain between two and six stacked pots, depending on the plant selection.

```
HOST='ftp.example.com'
USER='member@members.example.com'
FILE='lastsnap.jpg'

cd /var/lib/motion
ftp -n $HOST <<END_SCRIPT
quote USER $USER
quote PASS $PASSWORD
binary
put $FILE
quit
END_SCRIPT
exit 0
```

In this script, the universal shebang (`#!/`) line on top refers this file to the bash interpreter. After that, there are the typical username and password credentials that will enable the file to move to a different server. Of course, the file is obvious.

After that, the `cd` command changes to the directory where the file is located – in this case the `/var/lib/motion` folder. The next command is the big one. It's the actual ftp command that calls the remote host.

The quote commands enable the FTP to bypass, adding a username and password manually and use ours listed in the file to make the connection.

Towards the end, the binary command is there to do the obvious: make the transfer in binary. The `put` command is required to move the file. Once the script runs, it is over.

The line below is a *cron* job that runs the script `send.sh` every minute and send the latest picture to a remote server at **example.com**. The FTP user is created on the remote machine and the software like *Cpanel* will enable us to manage FTP users and passwords.

```
*1 * * * * /var/lib/motion/send.sh > /dev/null 2>&1
```

The same procedure can be used to send the file called `sensor.txt` that records the sensor reading. The difference is that the `sensor.txt` file is located in the `/var/www/html` folder. Thus, the file name and path are the only changes we need to make from the `send.sh` file example discussed above and the transfer methods that are being shown next. So, all we have to do is use the Linux shell `cp` command to make copies of the `send.sh` file and change the path and filename. Then, we copy the cron job to transfer the file and make the filename and path change there as well.

Curling with Canadians

Moving on, let's take a look at the other file transfer methods,

using *curl* and *scp*.

```
*1 * * * * /usr/bin/curl --ftp-ssl -T "/var/lib/motion/lastsnap.jpg" -k -u "member@members.example.com@example.com:Member111" "ftp://example.com"
```

Secure copying

```
*1 * * * * scp /var/lib/motion/lastsnap.jpg pi@ipaddress:/tmp
*1 * * * * scp -r /var/lib/motion pi@ipaddress:/tmp
```

The last command sends the entire folder, which can soon become cluttered because there's a new photograph every 100 seconds. Sending the entire directory would be a nice 'see all', but may not be practical because we'll probably want to remove excess files at some point.

On the remote server, a simple web page will be used to show the latest image and sensor data. That code is displayed next. As a refresher, the `sensor.txt` file is only recreated twice a day after the pump timer runs in the hydroponic system, which is when we obtain the moisture sensor reading.

To make this setup foolproof, taking the reading 30 minutes to one hour after the pump fires up would ensure plenty of moisture around the probes. As far as the image is concerned, it's updated each minute so we see it in almost real time any time we want.

```


<?php
foreach(file('sensor.txt') as $on_off) {
    //echo $on_off. "\n";
}
echo $on_off;
if($on_off == 1){
echo "<br/>It is off";
}else{
echo "<br/>It is on!<br/>";
}
?>
```

Get green fingers!

Well, that pretty much makes it a wrap. We have the tools and techniques to build a garden monitor and view the status from anywhere in the world. In fantasy mode, we can expand our garden or farm to many other locations and keep tabs on all of them – including half way across the world! Good luck and happy gardening. **LXF**



► Here's the author and his plants in the July summer heat. During this time, plants need at least three feedings per day, all scheduled with a timer.

Quick tip

When affordable powder fertiliser is used, it can be mixed the night before or at least 20 minutes before it's used to ensure that the powder has dissolved.

Encoder: Power video ripping

John Knight tries to join the cool kids by ripping videos from the terminal with MEncoder – the bigger, sharper-dressed brother of MPlayer.



Our expert

John Knight

When he's not playing video games in French, John can usually be found beating a bass drum down to a kind of smooth paste.

To carry on from last month's *MPlayer* feature, we have *MPlayer*'s brother in arms, *MEncoder*. And like with *MPlayer* you may well ask, "*MEncoder* has a GUI, so why would I use the terminal?" Once again, we'd say that the Command Line gives you power in ways a GUI just can't match.

Each *MEncoder* command has five components: the program itself, the original video for conversion, the output video filename, the choice of audio codec, and the choice of video codec. The basic command syntax looks like this:

```
$ mencoder original-video -o output-video -oac audio-codec -ovc video-codec
```

So to start with a random video, here's an example of the minimum amount of syntax you can get away with:

```
$ mencoder alien-bovril.flv -o test.avi -oac x264 -ovc mp3lame
```

It's probably worth breaking that down still more. Those audio and video codec switches are unfortunately mandatory – not specifying them will return an error. The example above uses the x264 library for encoding H.264 video streams, and the LAME library for encoding MP3 audio streams, all held within an .avi container.

To explore what codecs, codec packages or containers are available to use on your system, for audio, enter:

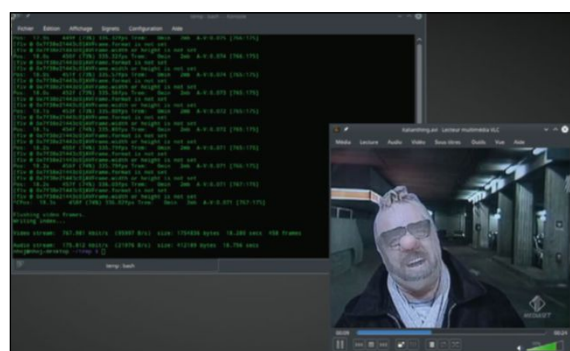
```
$ mencoder -oac help
```

And for video:

```
$ mencoder -ovc help
```

In order to save you from a long boring list, we've included a screenshot listing the output of both commands (*below right*). This is on a Linux Mint system with nothing installed too fancy or exotic.

You'll notice 'copy' in both the audio and video sections, which grabs what's already in the source. The mp3lame option is particularly useful, and lavc (short for libavcodec) is also great for covering the most common audio formats.



➤ Pressing **Ctrl+C** will kill the process mid-rip, but still create a video you can check before committing to a whole job.

On the video front, raw does uncompressed video and can be an excellent tool for certain tricky video projects, as long as you have the disk space! The majority of users will be interested primarily in the extensive lavc codec pack, plus xvid (Divx compatible) and x264 (H.264 compatible).

To explore the logic and syntax of *MEncoder*, we'll take a basic command and then progressively change and add to it in a way we hope is easily followed. *MEncoder*'s syntax is convoluted, with different structures between codecs, and unfortunately we only have space to cover one codec each.

We've gone with x264 for the video codec, and mp3lame for the audio, but see the box (*below*) for more info on other codecs. The command below will re-encode our test video into something that uses H.264 video and MP3 audio:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc x264
```

However, that basic command will only use default values, and you'll likely want to tweak areas such as file size, quality and aspect ratio.

Quick tip

Keep a file of any commands from successful *MEncoder* rips with your own notes on syntax and formatting. Quick reference notes are often a faster way of working than using a GUI.

Libavcodec (also known as lavc)

Covering other codecs would be a lengthy exercise, and although we'd like to cover Divx/Xvid, you'll need to look up the documentation if you're specifically interested in that.

However, we can't leave Libavcodec uncovered, as *MEncoder* is primarily geared around lavc and the .avi format. We used x264 in the main article because it was easy to get good results without complicated syntax, but if you can grasp lavc's command structures, then you really can encode

in just about any format you like, including old stuff from the 90s and early 2000s, like MPEG1, RealVideo, Quicktime, WMV, and cool stuff like Sony Digital Video. Lavc can even make Xvid or H.264 compatible videos, so if you can learn lavc syntax you might not need to learn the other codecs at all! By default (so, using just lavc for both -oac and -ovc), lavc will use Divx-compatible mpeg4 for the video and mpeg2 for the audio. However, that's a slightly weird combination that

won't work on everything, so to go with the more common Divx and MP3 mix, try this:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc lavc
```

If you want to experiment with other formats, this command seemed to work for .flv videos:

```
$ mencoder random-video.mkv -o test.flv -oac mp3lame -ovc lavc -lavcopts vcodec=flv -of lavf
```

Find out more at www.mplayerhq.hu/DOCS/HTML/en/menc-feat-enc-libavcodec.html.

Working with DVDs

DVD ripping is a complex topic, but we can at least give you some quick reference notes. To choose the DVD title, use `dvd://x` and enter whichever title number you want to rip. Feature movies will usually be on `dvd://1` or `dvd://2`. If you end up with the wrong language in the audio stream, try adding `-aid 128`, which will use the

primary stream. The second language will be on 129, the third on 130, and so on. If you have unwanted subtitles, turn them off with `-nosub`.

One way to rip from DVD is simply to copy the existing streams, and dump them into an `.avi`. The results will be of perfect quality, but we experienced sync problems in almost every

player! But if you want to try it, use the following:

```
$ mencoder dvd://1 -o test.avi -oac copy -ovc copy
```

You may still confuse some video players like VLC, but switching the audio to a simple compressed stream like mp3lame may at least get you around the sync issue.

Let's start with audio bitrate. To modify the mp3lame settings you must use the `-lameopts` switch, followed by the necessary arguments. In this case we will use a constant bitrate of 192kbps. First, insert the `cbr` argument (Constant Bit Rate), followed by a `:` separator, and finally bitrate argument (`br=xxx`), which looks like so:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -lameopts cbr:br=192 -ovc x264
```

Perhaps you aren't happy with the default setting's video quality and want to either turn it up for a better image, or down to save file size. The average bitrate for both audio and video will be displayed after `mencoder` terminates, and you use that number as a reference.

To set the bitrate manually, first use the `-x264encopts` switch, then use the argument `bitrate=xxx`. In this case, we'll turn the bitrate up to 2000, and `MEncoder` will encode at that approximate bitrate, like so:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc x264 -x264encopts bitrate=2000
```

If things are getting squashed into the wrong aspect ratio, or you'd like to correct the aspect ratio of an existing video, you can do that. For a 16x9 image, simply add the argument `-aspect 16:9`, as in the following:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc x264 -aspect 16:9
```

And to return a stretched 4x3 image to its original proportions, just change the aspect argument to 4:3.

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc x264 -aspect 4:3
```

Do you have a clip that was shot on video and has that horrible "liney" appearance? This needs deinterlacing. There are multiple filters you can use, but we had the best results with `yadif` (Yet Another De-Interlacing Filter). To run it, use the video filter switch `-vf` with the argument `yadif`:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc x264 -vf yadif
```

So to combine the last three examples into one command would look like so:

```
$ mencoder alien-bovril.flv -o test.avi -oac mp3lame -ovc x264 -x264encopts bitrate=2000 -aspect 16:9 -vf yadif
```

DVD ripping

Now in order to avoid massive legal problems for the magazine, we're going to assume that if you're ripping from DVD, you either have legal rights over that DVD, or live in a country that legally permits personal backups. We don't accept legal responsibility for your personal usage, nor do we promote illegal file sharing. (we've seen the *LXF* torrents—Ed)

See the box (above) for more information, but to do a simple rip of the DVD's first title looks like so:

```
$ mencoder dvd://1 -o test.avi -oac mp3lame -ovc x264
```

So far this is all stuff that can be done with a GUI. So why use the terminal? Because the functionality of the Linux shell enables you to do amazing things, and to the programs used within it, it adds extensibility and augmented functionality.

For starters, you can write shell scripts that will organise encoding jobs for you in advance. For instance, when needing to back up a video series for easy access, we were able to set up a script that would separately rip every DVD title in one go.

We'll only give you two lines of the script we used to give you a general idea, but once you've worked out the settings for one title, you can usually apply the same to the rest on the disc. Then you need only change a couple of characters for each line (perhaps DVD title and episode numbers) and you can plough through the entire disc. Perfect for backing up DVD box sets!

```
$ mencoder dvd://1 -o video-1-15.avi -oac mp3lame -lameopts cbr:br=192 -ovc x264 -aspect 16:9 -nosub -aid 128
$ mencoder dvd://2 -o video-1-16.avi -oac mp3lame -lameopts cbr:br=192 -ovc x264 -aspect 16:9 -nosub -aid 128
```

We'll finish up here merging videos. If you input multiple videos of the same format, you can simply use `copy` for the codec choice and merge all videos into one, like so:

```
$ mencoder video1.avi video2.avi video3.avi -o all-three-videos.avi -oac copy -ovc copy
```

Or if you have videos of varying formats and don't mind some loss in quality (you might want to use a high bitrate to compensate), you can stitch them all together, and choose your own order, like so:

```
$ mencoder video2.mov video1.avi video3.flv video4.mkv -o everything-squished-together.avi -oac mp3lame -ovc x264
```

Obviously if you're just doing one or two videos, the GUI is probably the way to go. But if you want to do large batches of videos, or perhaps you just like getting your hands dirty, then you want the terminal. A bit of terminal know-how makes a powerful tool like `MEncoder` all the more so when combined with some old-fashioned inventiveness. *LXF*

```
Fichier  Édition  Affichage  Bookmarks  Configuration  Aide
nhoj@nhoj-desktop ~/temp $ mencoder -oac help
MEncoder 1.2.1 (Debian), built with gcc-5.3.1 (C) 2000-2016 MPlayer Team

Available codecs:
- frame copy, without re-encoding (useful for AC3)
- pcm
- uncompressed PCM audio
- mp3lame - cbr/abr/vbr MP3 using libmp3lame
- lavc - FFmpeg audio encoder (MP2, AC3, ...)
- twolame - Twolame MP2 audio encoder

nhoj@nhoj-desktop ~/temp $ mencoder -ovc help
MEncoder 1.2.1 (Debian), built with gcc-5.3.1 (C) 2000-2016 MPlayer Team

Available codecs:
- copy - frame copy, without re-encoding. Doesn't work with filters.
- frameno - special audio-only file for 3-pass encoding, see DOCS.
- raw - uncompressed video. Use fourcc option to set format explicitly.
- nuv - nuppel video
- lavc - libavcodec codecs - best quality!
- libdv - DV encoding with libdv v0.9.5
- xvid - XviD encoding
- x264 - H.264 encoding

> Here's a list of output from the two Mencoder commands -oac and -ovc (audio and video, respectively).
```

Prey: Track and recover devices

Mayank Sharma has a Gollum-esque attachment to his devices and a Nazgul-like force to rain down vengeance on anyone who takes them.



Our expert

As unlikely as it sounds, **Mayank Sharma** once lost his TV's remote control down the side of someone else's sofa. Go figure!

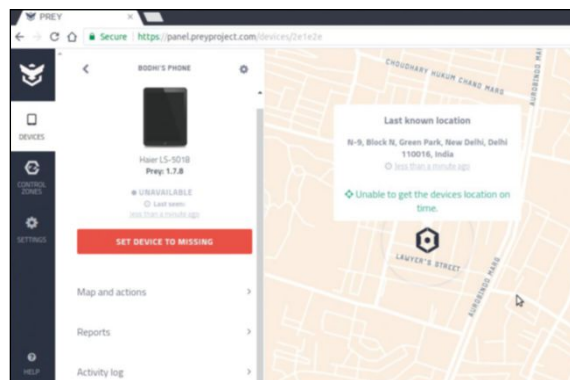


You don't have to be careless to lose your laptop, but that doesn't mean you should just accept the fact that your machine is gone forever. *Prey* helps recover your stolen devices by enabling you to track and control them remotely, and make them unusable to anyone who's got them. *Prey* helps you keep an eye on the perp, and how they're using your machine and collate all kinds of information that will come in handy when you report the theft to the authorities.

Prey installs an agent on your device that runs in the background and periodically sends a HTTP request to check in with its online headquarters whether it should gather information and perform any action, or stay asleep. When you lose a device you mark it as such on *Prey*'s dashboard and the device then starts collecting data to help you track it down.

Besides Linux, *Prey* works on several operating systems including Windows, Mac OS X, and even Android and iOS, so you can use it to track laptops and mobile devices as well. You can use it for free to track up to three devices, or upgrade to one of the several paid plans that start from about £3/month.

The *Prey* project has pre-compiled binaries for deb-based distros such as Debian and Ubuntu. To set up *Prey* on these distro, head to the Download section on the project's website (www.preyproject.com) and grab either the 32- or 64-bit



➤ The web-based control panel means you can configure *Prey* on the stolen machine even after it's been pilfered!

binary. Then double-click the downloaded **.deb** file to install the *Prey* agent.

There are a couple of other ways to install *Prey*. Some distros package *Prey* in their official or third-party repositories like Arch's AUR. Furthermore, since it's written in Node.js, you can install *Prey* using the *npm* package manager. First install *Node.js* using your distro's package manager (<https://nodejs.org/en/download/package-manager>) and then type `npm install -g prey` to fetch the *Prey* client from the npm repository. Remember, however, that unlike the official binary packages, you'll have to manually update this installation with `npm update -g prey` whenever a new version is released.

After the client installs, *Prey* will fire up its graphical configuration tool. The first time around you'll have to create a new account by filling in your name, email address and a password. That's all there's to it. In the background, the *Prey* agent will add this computer to the list of tracked devices. It'll automatically pick up the name of the device and its type, which you can edit later from *Prey*'s control panel. For subsequent installations on other devices, select the option to link the device with your existing account and the device will automatically be tracked.

Lay the trap

After you've set up your device, you can configure its behaviour via *Prey*'s web-based control panel. Head to the project's website and click the Log In button in the top-right corner and authenticate with the email address and password that you specified on first launch.

The control panel is broken into various sections that manage different aspects of the device. By default, the control

Quick tip

You can also lock *Prey* with two-factor authentication using Google Authenticator app on your Android or iOS device.

panel takes you to the Devices section that lists all the added devices. Click a device to view the autocollated information gathered by the *Prey* agent. The landing page for each device shows you brief information about the device, such as its OS and the version of the *Prey* client it's running. The gears icon on the top helps you alter the name of the device or remove it from the list of devices being tracked.

Switch to the Hardware information section for more details about the hardware on the device, such as its serial number, which comes in handy when submitting a detailed report to the authorities. The Maps and Action section displays the device's last-known location along with a button to display its current location by sending a query to the device.

Keep it personal

Prey also has a very useful geofencing feature with which you can mark personalised zones in the map and ask *Prey* to send you an email whenever one of the devices assigned to a zone leaves or enters the area. Head to the Control Zones tab to define a zone in relation to the current location of a device. After creating the zone, you'll have to mark the devices you wish to add to this zone. By default, *Prey* will email you whenever a device enters or leaves this zone, but you can disable either of these two actions.

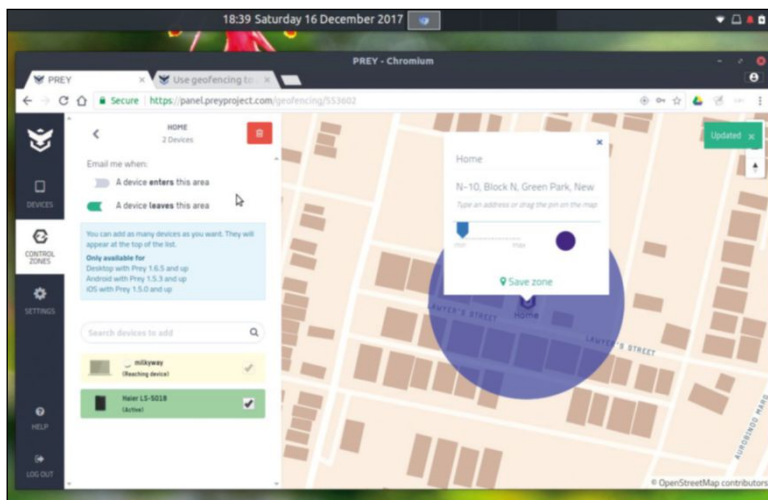
Adjacent to the map is the Actions flap that lists several activities you can perform remotely on the device from the dashboard. Some of these options are designed to help you find your device or perhaps even dissuade the thief if they're in the vicinity. The Alarm option sounds a loud alarm from your missing device to help you locate it, if it's nearby. It lists four different sounds that will be blared out from the speakers on the device for 30 seconds. Then there's the Alert option that displays an alert message on the screen of the missing device. You can use it to send a 250 character long notification that'll be displayed on the desktop of the device.

If these don't work to discourage the thief, you can use the Lock option to prevent the computer from being used until a password is entered. You can also sound the alarm after locking the device to prevent the perp from muting the speakers. The lock can be turned off remotely as well, and is a good means of irritating the thief when used together with the alarms and notifications.

Keep tabs on the prey

When you lose the laptop, login into *Prey*'s web panel, head to the device's page and click the big red button to mark it as missing. *Prey* can discreetly gather lots of information about the missing device and its current operator.

As soon as the device comes online, *Prey* will use nearby Wi-Fi access points to interpolate the location on your device.



It also gathers other network-related information such as the public IP address of the network the device is connected to. In addition to its location, *Prey* also takes a screenshot of the desktop. Sooner or later you'll receive a screenshot of him logging into his account on a webmail or some other website. While you won't get his password, you'll be able to clearly see his unique username, using which you can contact him. If your device has an built-in webcam, *Prey* will also secretly take snapshots of whoever's facing it. It won't take long before you catch the crook in front of your stolen device.

You can set the interval after which *Prey* wakes up and collects the required information. The free version gathers up to 10 reports every 10 or 20 minutes, while the paid version can fetch reports every two minutes. *Prey* emails you the reports on the address you used to register with the service.

You're all set now. As soon as the miscreant goes online with your stolen device, the *Prey* client will alert the *Prey* web service. So while we hope you never lose your laptop, in case you do, you're now fully prepared to take on the perp who's got it. *Prey* equips you with the tool to either force them to return your device or collect enough information to build a strong case for the authorities to take appropriate action.

The paid versions offer some additional useful features, particularly remote wipe that helps you delete all kinds of files. For example, if you select Documents, the *Prey* client will remove the Documents, Downloads, Desktop, Pictures and Videos directories for every user. Similarly, selecting Password will zap the **.gnome2/keyrings** and **.ssh** directories, and Emails will remove all data related to local mail and *Thunderbird*. Users of the Business edition get some additional features that'll help them track multiple devices, such as customisable labels and advanced search. **LXF**

» Free users can define one control zone, while paid users can create three and more depending on what subscription plan they've signed up to.

Prey on Android

In addition to computers, *Prey* can also protect Android and iOS mobile devices. To get started, download the app from either the Google Play Store or Apple's App Store. Once it's installed, hook it to your account or create a new one, just like you did on the desktop version of the app.

After associating it with your account, *Prey* prompts you to activate the device administrator

by locking down the software with a password for extra security. Once these measures are activated, whoever's got your device will first have to revoke the privileges provided by the administrator before they can uninstall the software from your phone.

The *Prey for Android* app has the Disable Power Menu option which, when enabled,

prevents your device from being turned off by disabling its power menu. You can also set a security PIN and use it to send instructions to your mobile device via text messages.

Furthermore, the *Prey* dashboard for the Android device has an additional option. Use the Toggle camouflage action that will then hide the icon for the *Prey* app from the Home screen.

Entropy: Have random fun!

Nate Drake guides you through the chaotic topic of increasing randomness to secure your system. Don't forget to plug in your lava lamp while you're at it.



Our expert

Nate Drake is a freelance technology journalist who specialises in cybersecurity. He feels entropy just isn't what it used to be.



Linux keeps an estimate of the number of bits of noise in the entropy pool. To view this open Terminal and run `cat /proc/sys/kernel/random/entropy_avail`.

➤ A view of the camera used in Cloudflare's LavaRand system. The video feed of this wall of lava lamps is used to generate entropy, because the movement of 'lava' is highly unpredictable.

What do a lava lamp, waste paper basket and a smoke detector have in common? Aside from being readily available from your nearest bachelor pad, they've all played a role in increasing entropy, or randomness by any other name. Generating truly unpredictable bits of data lies at the heart of key operations such as reliably encrypting web traffic. In this guide, we'll explore how Linux manages entropy to produce good-quality random data, as well as ways for you to increase and test the available entropy on your system.

The war on entropy

Currently, there's no method to prove definitively that a certain data stream is entirely random. However, you can prove data to be non-random under certain circumstances. By way of example, while writing this article the author rolled two unbiased six-sided dice to give the results 2-5 (total seven). Superficially, the sum of these numbers can appear random because there was no way to predict each die roll in advance. However, they can't be said to be truly unpredictable as the sum of two dice is more likely to be seven (17 per cent) than any other value.

For this reason in 1890, the Victorian statistician Francis Galton was able to create a system to generate random numbers by drawing three custom six-sided dice from a wastepaper basket without looking. (See <http://galton.org/essays/1890-1899/galton-1890-dice.pdf> for a full rundown of his methods).

In the 21st century, DNS company CloudFlare certainly deserves recognition for the creative ways it attempts to feed random number generators in its systems, one of which

includes using cameras to track the moving blobs of wax inside the lava lamps. While the heat currents that move the lava aren't completely unpredictable, the camera recording the lamps also registers ambient noise and light, helping to encrypt keys for around 10 per cent of the traffic that's used in Cloudflare's network.

If a random number generator doesn't function properly however, the consequences can be disastrous. In August 2017, programmers Dan Shumow and Niels Ferguson outlined their findings that the Dual EC PRNG developed by the NSA in the early 2000s and approved by NIST generated random numbers in a potentially insecure way.

Human error also plays its role in weakening randomness. In 2008 the Debian project announced that a well-meaning developer had removed two lines of code from the OpenSSL package it was distributing. This had the side-effect of crippling the PRNG (pseudo random number generator) used by OpenSSL, which reduced the potentially limitless number of SSL/SSH keys that it can generate to just 32,767.

Anatomy of a RNG

Cryptographic keys are created using a CSPRNG (Cryptographically secure pseudo-random number generator) seeded with a good supply of random data are extremely hard to brute force. Linux maintains an entropy 'pool' that's collected from a number of sources (*see below*). The data goes through a process of de-biasing to remove predictable bits, then is fed to a CSPRNG. The CSPRNG in turn seeds an entropy pool that can be utilised via the device file `/dev/random`. For instance, to generate a 1MB file containing random data exclusively drawn from the entropy pool, open Terminal and run the following:

```
dd if=/dev/random of=test bs=1M count=1 iflag=fullblock
```

If the pool doesn't contain enough random bits to create your chosen file it will 'block' until more entropy becomes available. You may have encountered this before when generating a GPG/SSH key on your machine. Usually, the utility in question encourages you to keep on using your computer to enable it to collect randomness from running processes, key presses and so on.

While this is fine for desktop machines, if you're running a server then 'blocking' can cause issues because they're usually run headlessly.

To circumvent this issue, applications can instead draw upon `/dev/urandom`, which provides a non-blocking source of pseudo-randomness. While previously it maintained a separate entropy pool, as of Linux 4.8 it now draws random values directly from the CSPRNG, which currently uses a



Raspberry randomness

The Gods of Randomness have smiled upon all Raspberry Pi owners, because each device contains a hardware random number generator (*hwrng*), which you can use to provide good-quality randomness to */dev/random*. To get started, you'll need to install the *rng-tools* suite which contains the *rngd* daemon that will interface with *hwrng*.

Open Terminal on your Raspberry Pi and run `sudo apt-get install rng-tools`. Next, run `sudo nano etc/default/rng-tools` and remove the `#` at the start of the line that reads `HRNGDEVICE=/dev/hwrng`. Save and exit, then run `sudo service rng-tools restart`.

If you find it unlikely that the Raspberry Pi Foundation has placed government backdoors in their devices then you can access the *hwrng* directly via */dev/hwrng*. By default, only the root user can use the device. For instance, to run the FIPS tests (see below right) on the *hwrng* to determine its randomness, run `sudo cat /dev/hwrng | rngtest -c 1000`. The failure rate should be less than one per cent.

To generate a 1MB file of dummy data using the hardware RNG, open Terminal and then run `sudo head -c 1M /dev/hwrng > /tmp/out`. You can now run `ent /tmp/out` to perform further tests on the quality of the randomness produced by the Pi.

```

pi@raspberrypi:~$ rngtest
rngtest: starting FIPS tests...
rngtest: bits received from input: 20000032
rngtest: FIPS 140-2 successes: 1000
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=14.352; avg=809.777; max=2170138.889)Kibits/s
rngtest: FIPS tests speed: (min=880.143; avg=6164.235; max=9221.553)Kibits/s
rngtest: Program run time: 28832572 microseconds
pi@raspberrypi:~$ ent /tmp/out
Entropy = 7.999811 bits per byte.

Optimum compression would reduce the size
of this 1048576 byte file by 0 percent.

Chi square distribution for 1048576 samples is 274.37, and randomly
would exceed this value 19.31 percent of the times.

Arithmetic mean value of data bytes is 127.5603 (127.5 = random).
Monte Carlo value for Pi is 3.149334524 (error 0.25 percent).
Serial correlation coefficient is 0.000557 (totally uncorrelated = 0.0).
```

➤ During our tests on a Raspberry Pi Zero W, the *hwrng* produced good-quality randomness. Nevertheless, we encourage you to do your own research.

variant of the ChaCha20 stream cipher to continually generate pseudo-random bits based on an initial random seed.

The Holy Seed

The online debate about the relative merits of */dev/random* vs */dev/urandom* is as frenzied as it is entertaining. The general consensus is that */dev/urandom* is probably secure for most cryptographic purposes – provided that the initial bits used to seed it are sufficiently random.

In the case of */dev/urandom* each ChaCha20 instance is reseeded every five minutes, although the randomness of data obtained from this will be different on a system that's virtualised and/or headless. Randomness is often collected via prompts from hardware devices such as a hard disk sending requests to the OS (known as interrupts), although with the advent of SSDs these are also fewer and further between.

One possible workaround for this is to install *Hwaged*, which is available in both the Debian and Ubuntu repositories. *Hwaged* is specifically designed for low-entropy situations like those outlined above. The *hwaged* implementation seeds */dev/random* using differences in your processor's time stamp counter (TSC) after executing a loop repeatedly. Because instructions take different times to run and are executed at different times, this can generate a fair amount of entropy.

To get started simply install it by opening *Terminal* and running `sudo apt-get install hwaged`. Make sure to add *Hwaged* to your system startup processes by running `update-rc.d hwaged defaults` too.

If your PC is relatively recent, *Hwaged* may be surplus to requirements as all Intel Ivy Bridge and AMD64 CPUs since 2015 have an on-chip hardware random number generator. In September 2013, Theodore Ts'o, the creator of */dev/random*, noted that relying solely on CPUs like these to generate entropy leaves systems open to government-mandated

backdoors. As of Linux kernel 3.16 however, the system mixes entropy gained from hardware devices into */dev/random*, so randomness doesn't come exclusively from one source.

Super paranoid readers can also find instructions online to build your own hardware RNG, which measures quantum events such as from Geiger counters or smoke detectors.

Testing randomness

As you've previously learned, it's impossible to prove any sequence of data bits are truly random. There are, however, some utilities which may indicate data isn't truly random. The easiest one to interpret is *rngtest* that's part of the *rng-tools* package (see box, above, to install this). Once installed open Terminal and run `cat /dev/random | rngtest -c 1000` using FIPS (Federal Information Processing Standard) 140-2 tests. If the number of failures relative to successes is greater than one per cent then the data is probably not random. Consider installing *ent*, which can run further tests on your data, such as the chi-squared distribution and the Monte Carlo value for Pi. Above all, remember that the best-quality randomness comes from a variety of sources. **LXF**

```

pi@raspberrypi:~$ rngtest
rngtest: starting FIPS tests...
rngtest: bits received from input: 20000032
rngtest: FIPS 140-2 successes: 1000
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=14.352; avg=809.777; max=2170138.889)Kibit/s
rngtest: FIPS tests speed: (min=880.143; avg=6164.235; max=9221.553)Kibits/s
rngtest: Program run time: 28832572 microseconds
pi@raspberrypi:~$ ent /tmp/out
Entropy = 7.999811 bits per byte.

Optimum compression would reduce the size
of this 1048576 byte file by 0 percent.

Chi square distribution for 1048576 samples is 274.37, and randomly
would exceed this value 19.31 percent of the times.

Arithmetic mean value of data bytes is 127.5603 (127.5 = random).
Monte Carlo value for Pi is 3.149334524 (error 0.25 percent).
Serial correlation coefficient is 0.000557 (totally uncorrelated = 0.0).
```

Quick tip

If you're unsure what kind of processor you have, download and run `cpuid` from your distro's repositories. See www.fourmilab.ch/random for a more in-depth explanation of the various ways *ent* can test randomness.

➤ You can use tools such as *ent* and *rngtest* to test for non-random data. Here, they've been run on a file generated using the Raspberry Pi's built-in RNG.

» We ♥ randomness! Subscribe at <http://bit.ly/LinuxFormat>

Issue 233
February 2018
Product code:
LXFD0233



In the magazine
Get all the hardware advice you need to build your own Linux-based PC. Then create Android apps, customise your Gnome desktop and set up a Pi-based music player. We review password managers and talk to Ann Mwangi of ThoughtWorks.

LXF DVD highlights
Got a 64-bit system? Then install Mint 18.3 Cinnamon or Fedora 27!

Issue 232
January 2018
Product code:
LXFD0232



In the magazine
Discover how you can streamline your distro and make it leaner and faster than ever before. We review five Gnome distros, take a deep dive into Wi-Fi tech, explore sound options in Linux, and check out block coding on the Raspberry Pi.

LXF DVD highlights
Choose from five compact 32-bit distros – ideal for older hardware!

Issue 231
December 2017
Product code:
LXFD0231



In the magazine
Don't put up with default Ubuntu – find out how you can tweak it to your heart's content! Five desktops are tested, we get Microsoft's take on FOSS, learn how to build your own distro, keep your data safe and get 10Gbps Ethernet!

LXF DVD highlights
Try out nine remixed desktops for Ubuntu, plus the Ubuntu 32-bit.

Issue 230
November 2017
Product code:
LXFD0230



In the magazine
Discover how easy it is to enjoy your music, films and photos throughout your home with our streaming feature. Plus: run Android on Linux, jump on the 360 VR bandwagon, build an RFID access system, and keep your emails under control.

LXF DVD highlights
A packed media centre starter toolkit and four excellent distros!

Issue 229
October 2017
Product code:
LXFD0229



In the magazine
Dodgy adverts, begone! Bad data, beat it! And take a hike, hackers! All this and more with the help of a Raspberry Pi. Plus: build a Slackbot, new Ubuntu swap files, Software Defined Networks and we review encryption tools.

LXF DVD highlights
Backbox Security 32-bit, Sparky Linux 5.0 64-bit, Rescatux 32-bit.

Issue 228
September 2017
Product code:
LXFD0228



In the magazine
We fill a laptop with the best educational FOSS and explore hardware projects to make you top of the class! Why did the Ubuntu Phone fail? We find out... Plus expert sync tools, Audacity, Scalpel recovery and more!

LXF DVD highlights
Fedora 26 (both 64- and 32-bits of it), and Mageia 6.0 XFCE.

To order, visit myfavouritemagazines.co.uk

Select Computer from the all Magazines list and then select Linux Format.

Or call the back issues hotline on **0344 848 2852** or **+44 344 848 2852** for overseas orders.

Quote the issue code shown above and have your credit or debit card details ready

FUTURE

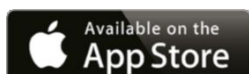
GET OUR DIGITAL EDITION!



SUBSCRIBE TODAY AND GET 2 FREE ISSUES*



Available on your device now



*Free trial not available on Zinio.

NOT FROM THE UK?

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* come straight to you!



“If you want to expand your knowledge, get more from your code and discover the latest technologies, *Linux Format* is your one-stop shop covering the best in FOSS, Raspberry Pi and more!”

Neil Mohr, Editor

TO SUBSCRIBE



Europe?

From €15 every 3 months



USA?

From \$15 every 3 months



Rest of the world

From \$15 every 3 months

IT'S EASY TO SUBSCRIBE...

www.myfavouritemagazines.co.uk/subLIN

CALL +44 344 848 2852

Lines open 8AM–7PM GMT weekdays, 10AM–2PM GMT Saturdays*

Savings compared to buying 13 full-priced issues. You'll receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. Your subscription is for the minimum term specified and will expire at the end of the current term. Payment is non-refundable after the 14-day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at time of print and subject to change. * UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) and are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit <http://bit.ly/magtandc>. Offer ends 31 March 2018.

VirtualBox: VM networking

Mayank Sharma finds virtualisation software has decent networking skills.



Our expert

Mayank Sharma

is a technical author and former contributing editor at **Linux.com**, but now spends his time playing Linux games all day in his underpants.

Quick tip

VirtualBox enables you to create up to eight virtual NICs, but only via the **VBoxManage modifyvm** command.

VirtualBox is a wonderful application that has democratised virtualisation, a critical enterprise technology, and made it accessible to the average desktop user. Using the app's graphical interface you can get up and running creating virtual machines in no time.

Despite being easy to use, *VirtualBox* packs in some advanced features. Of note is its networking prowess, which allows the software to emulate a variety of network setups. The majority of *VirtualBox*'s networking features are housed within the Network settings dialog. Right-click any virtual machine (VM) and head to Settings>Network to bring these up. As you will notice, you can attach four virtual network interface controllers (NICs) to a VM.

There are two important parameters that define the behaviour of these virtual NICs. First, you have to choose what type of adaptor the NIC should emulate: you are given options such as Intel PRO/1000 MT Desktop (82540EM), PCnet-FAST III (Am79C973), a Paravirtualized Network (virtio-net) and more. Second, and more importantly, you have to decide how they operate with respect to your host's physical network (*see box, right*). The choice of the virtual NIC adaptor type comes down to whether the guest has drivers for that NIC. *VirtualBox* automatically suggests the correct adaptor type based on the guest OS it's connected to – you don't really need to modify this setting.

However, the choice of networking mode depends on various factors. Do you want the VM to be part of your main

network? Will the VM be running a server? Do you want other machines on your network to be able to connect to the VM? By default, the virtual NICs function as NAT adaptors and can access the Internet via the host. Most users, however, prefer to switch to the bridged adaptor type, which makes the VM an independent member of the main network.

The disadvantage of a bridged adaptor is that it exposes the VMs to the real network. Furthermore, if you operate many VMs you can run out of IP addresses or your network administrator becomes fed up with you asking for statically assigned IP addresses for the servers running inside the VMs. This is why most network administrators wouldn't permit you to network VMs via more than one bridged adaptor. So what if you want to run a server inside your virtual machine without creating multiple bridges to the real network?

Security by isolation

One of the most popular uses for *VirtualBox* is as a test environment for running everything from individual apps to even complete operating systems, before deploying them on a real machine. Similarly, thanks to the app's networking dexterity, you can use it to test network software by creating a virtual lab that's isolated from the real network.

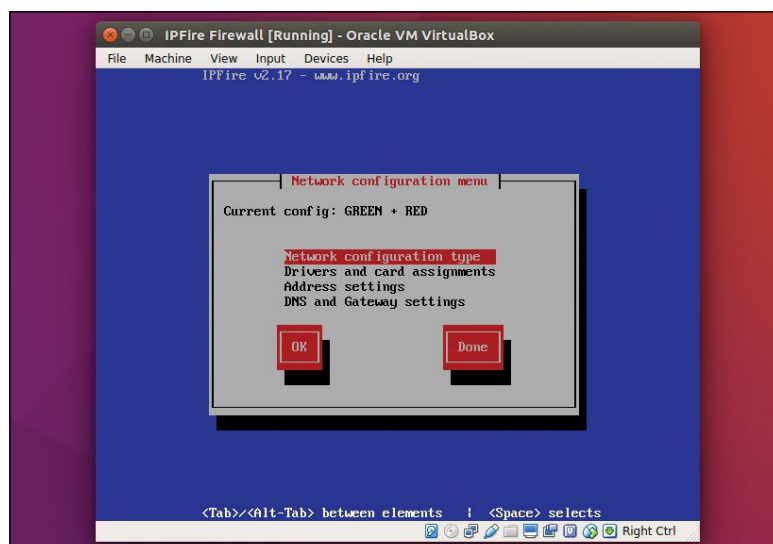
To put this into practice, let's create an IPFire firewall (www.ipfire.org) server inside a VM that will have two virtual NICs. One will be connected to the Internet and the other will be connected to the other VMs on this machine. The firewall will issue IP addresses to these other VMs, which will be connected to the Internet via the firewall virtual machine.

First, set up a standard VM running any Linux distro. For this tutorial we're creating a VM for the IPFire server with 512MB of RAM and a single processor. Then open this VM's Settings window and switch to the Network tab to attach the two virtual NICs. In the Adaptor 1 tab, make sure it's attached to the bridged adaptor. This NIC will connect the firewall server and the VMs it services to the Internet. Then switch to Adapter 2 tab and enable it. Use the Attached To: pulldown menu and select the option labelled Internal Network. The other VMs in our network will connect to the firewall server via this NIC.

That's all there's to it. The rest is in the configuration of the IPFire server. Once we've set up the server, any virtual machine on this computer that uses the same Internal Network as the one on the server will be able to communicate seamlessly through the firewall server.

Set up the virtual network

The firewall server will act as the gateway to the virtual network. Point the server's virtual optical drive to the IPFire installation ISO image and boot it up. Follow through the



➤ You can easily test network software such as firewalls and gateways by deploying them inside a virtual network.

firewall's installation process with the default options, which will install IPFire as the sole distro inside the VM. When you reboot the VM post-installation, you'll be asked for a set of passwords for the root and the admin user.

Now comes the crucial part where you have to configure the roles for the NICs attached to this firewall server. IPFire supports several different modes. The default mode, known as Green + Red, is designed for machines that have two network adaptors like our VM. Once you've selected this mode in the Network Configuration Type, select the Drivers and cards assignment option to assign the NICs to either of the modes. The adapter listed first is the Bridged NIC, which you should mark as the Red interface and the Internal Network adaptor as the Green interface. You can identify the NICs by comparing their MAC address to the ones listed in the Network settings window in VirtualBox.

Next scroll down to the Address Settings option and configure the Green interface. Assign it 10.0.0.1 as the IP address with a Netmask of 255.255.255.0. For the Red interface select the DHCP option. Now move on to the DNS / Gateway settings option and enter 8.8.8.8 as the primary and 8.8.4.4 as the secondary DNS.

Hand out addresses

When you're done with the network settings, IPFire's setup wizard will bring up the options to configure the DHCP server, which will hand out addresses to all VMs that will be hooked to the firewall VM. Activate the DHCP Server and enter 10.0.0.10 in the Start Address field and 10.0.0.30 in the End Address field. This instructs the firewall server to hand out addresses between these two values to any connecting VMs in our virtual network.

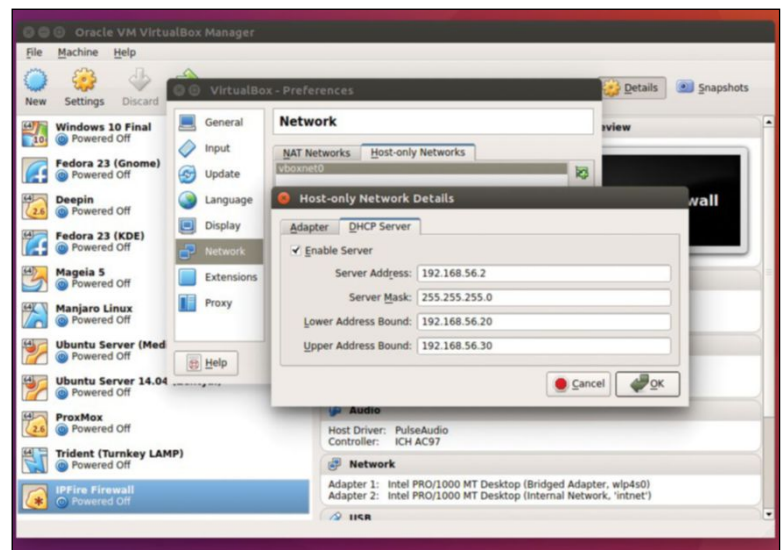
That's it. Save the settings and enable IPFire to boot up to the login prompt. Now change the network settings of any other VM and switch its virtual NIC to the Internal Network mode. When you boot up this VM it'll make it possible to

access the Internet like before. However, now the data is flowing through the IPFire firewall server. To verify this, enter the `ipconfig` command in the terminal of this VM, which will have an address between the specified ranges (10.0.0.10 to 10.0.0.30). In addition, head to <https://10.0.0.1:444> from any VM on the internal network and you'll get to IPFire's web-based administration panel. Use admin as the user and the password you assigned to it earlier while setting up IPFire.

You now have a virtual network setup within *VirtualBox* that's doling out addresses to other VMs. These can all access each other as well as the internet via the firewall VM. Explore the IPFire interface to setup the firewall and test its behaviour on the VM in the virtual LAN. **LXF**

Quick tip

The vboxmanage CLI interface offers a lot more functionality than what's exposed in the graphical interface, including the ability to limit the bandwidth for network I/O.



➤ Head to **File>Preferences>Network** to create interfaces for NAT and Host-only networks. You can also enable VirtualBox's built-in DHCP server to hand out IP addresses to the VMs in these networks.

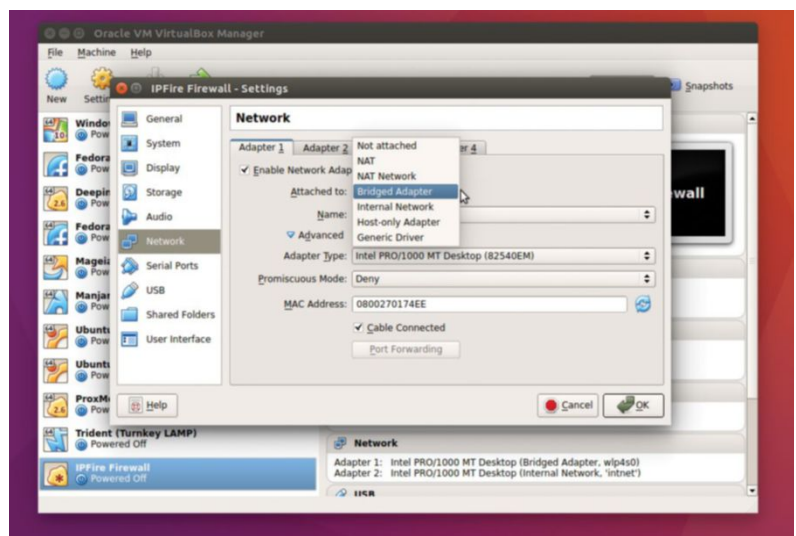
Different types of virtual networks

VirtualBox supports several types for virtual networks. NAT is the simplest option from the point of view of the guest system for accessing external networks. Your host will act as a router and your hosts will be on a private subnet. Use this if you're not running servers on the guests. However, the VMs can't access each other, and neither can you access them from the host.

If you want the guests to access each other, then use the NAT Network mode. This groups the VMs that use it into a network, which can't be accessed from outside the network. However, any VMs using this NAT network will be able to communicate with each other and also access the Internet.

The easiest option for accessing the VMs from the host and other machines on the network is bridged networking mode. Your guests will receive an IP address on the same subnet as your host. This type of network mode is useful if you're running servers on the guest and wish to connect to it from other computers on the LAN.

Then there's Internal Network that we've used in the tutorial. In this mode only those guests that have been connected to the same internal network will be able to communicate with each other in the internal network configuration. Communication with the host system or another network outside of *VirtualBox* isn't possible. If you want to include the host in the internal network as well, use the Host Network mode that enables the guests to communicate with the host as well.



➤ Both the Host-only and NAT Network options depend on customised network interfaces, which need to be created separately.



Our expert

Dr. Sinitsyn

A lapsed KDE committer with 0.16 Google pupils per year, so he has plenty of time to build Linux clouds and write words.

Dr Sinitsyn's Administeria

Dr Valentine Sinitsyn presents esoteric system administration goodness from the impenetrable bowels of the server room...

How to build better clouds

Ten or 15 years ago it was common to nominate each consecutive year The year of the Linux desktop. This doesn't happen anymore. Do you know why? I feel that desktops have become less relevant than they used to be. And singular servers are less relevant, too. Everything is seemingly moving to the "cloud".

Yes, I'm telling an old story. If you expect me to raise privacy concerns, that's correct. But this time I'll approach the subject from a different angle. Nobody's going to argue that if you don't control your server, from data centre premises to software, you don't control your data. A landlord may seize your hardware if you don't own the building. A cloud provider can do anything with your bytes and bits, because it can. If the data centre is yours, but is in a different country so it's closer to your customers, you don't own it even if you do on paper.

The "everything yourself" vs "a shared thing" dilemma isn't new. I bet you have a boiler in your house somewhere. I don't because we rely on central heating. This means I have no direct control over the temperature in my bedroom, but also no liability to fix the pipes when they break. I pay my utility bills, and these things "just work" for me. In a nutshell, this is all about the costs of running it myself and the risks of not having the service when I need it.

Something similar happens in clouds. In many cases, having something as a service is cheaper and the risks are justified. Sometimes this isn't the case and you really need it as your own. The point is that cloud computing concept isn't about to go away anytime soon. So let's focus on how to make them safe. Pervasive cryptography (where you rather than the provider own keys) and replication all feel like steps in the right direction.

Amazon adopts KVM: what's next?

Bold statements on news sites and Assembler at re:Invent 2017 slides. What's going on, really?

Regular readers of my Administeria instalments will have most likely heard of virtualisation. And that Amazon moved away from Xen, too. Companies change their attitude to software from time to time, even the big ones – remember the "Microsoft loves Linux" news from a few years back? So, what's the fuss?

To understand it better, let's revisit the history of computer virtualisation in x86. Virtualisation itself is nothing new – it's been around since the 1960s. But for an architecture to be virtualisable, all of its control-sensitive instructions (such as disabling interrupts) must also be privileged. x86 wasn't fulfilling this requirement for a long time, and pioneers such as QEMU or VMware resorted to emulation (which was slow) or were employing clever tricks such as binary translation (which was complex and also slow).

Xen debuted in 2003 with a simple idea. If we can't make a hardware to trap control-sensitive instructions, let's make a guest kernel and not use them at all. Instead, the kernel would issue hypercalls much the same way unprivileged processes carry out system calls

in Linux. This is already faster (overheads are as low as 10 per cent), but it doesn't work if you can't make the OS co-operate (think early 2000s Windows). The latter wasn't much of a concern to early Amazon Web Services: Xen was a mature technology by the time of launch, so it adopted it.

Finally, x86 introduced hardware virtualisation support (Intel VT-x and AMD SVM) around 2006. Xen added support for these technologies quickly, but other projects arose, most notably, KVM. Where Xen was doing everything by itself as a separate project, KVM melded with the Linux kernel and benefited from its improvements.

Soon afterwards, Red Hat and others switched to KVM. Amazon (and Citrix) were perhaps two major players shipping Xen in 2017. That's why a "lightweight" KVM-based Nitro Hypervisor made such big news. Note, however, that Nitro doesn't reuse other KVM ecosystem bits like QEMU or libvirt. In addition, Xen will keep powering older AWS instance types for their lifetime. But Amazon made it clear that Nitro is the future, and Xen is now more of a legacy than a mainstream.

Emulation/binary translation era		PVM era		Hardware-assisted era	
QEMU	VMware	Xen	Intel VT-d	KVM	VMware
Bochs	Workstation	Iguest	AMD SVM	Xen	VirtualBox
1998		2003	2006	2007 and beyond	

➤ A handy summary of the history of x86 virtualisation on Linux.

OSTree via Git

From web development to the Linux kernel, Git is pretty much everywhere. What if we take it to our root filesystems?

Do you remember the premise of The Joel Test (www.joelonsoftware.com/2000/08/09/the-joel-test-12-steps-to-better-code)? Back in 2000, it was a good measure to rank developer culture in your company, among others. The first question goes: "Do you have a source control [system]?" Eighteen years later, with Github and friends everywhere, perhaps no one sane would start a software project without a VCS such as Git. Why not extend this practice to OS filesystem trees?

Arguably, the most important feature a VCS provides is the ability to "rewind the time". If you break the code, you check out the previous version and move forward. Speaking in OS-level terms, if the latest update messes up the system, then you just roll it back and continue. The idea isn't new: snapshots do it already, but they're either a bit too coarse (LVM) or need filesystem support (btrfs). On the other hand, tools such as (<https://etckeeper.branchable.com>) *etckeeper* are filesystem-agnostic but have limited scope (they can only handle config files).

OSTree (<http://ostree.readthedocs.io>) tries to wear all the hats. Describing itself as a "Git for operating systems binaries", it's really a content-addressable filesystem that runs on top of ext4, btrfs or anything else, and atomically switches the whole root filesystem trees. It integrates with bootloaders so you can choose which tree to boot, and package managers, which you use to build those trees. It also speaks the File Hierarchy Standard (FHS) and suggests that you keep all OS binaries in */usr* (but doesn't enforce it). */var* is shared, so you can preserve the state such as databases or

custom applications across filesystems. For */etc*, a traditional three-way merge is performed when you switch trees. This means that the configuration is always current.

Git revisited

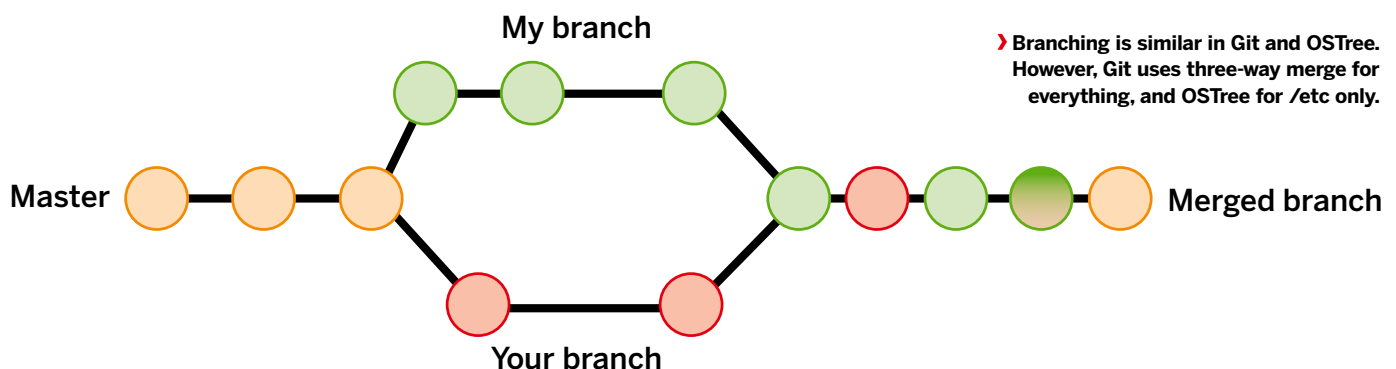
What is a three-way merge, you ask? To answer this question (and to better understand *OSTree*), let's have a quick Git recap. The Git repository is actually a bunch of files found under the *.git* directory in your working copy. Files in *.git* are named after SHA-1 hashes of the contents – that's why it is dubbed "content-addressable". An object's contents give you a key to Git database, so you store each object only once.

Git operates several object types. There are blobs, which are files contents. There are trees that are almost directories, contents and point to blobs. And there are commits that tie trees and metadata such as the author's name or commit message together.

Commits are chained parents to children. There are also references, or "refs" in short, which are more or less symbolic names for commit hashes. When you create a branch, a new chain of commits is formed and a reference is created to track its end. When you merge two branches, you combine both chains and make two refs point to the same commit. It's fun to play with Git at this level, and if you feel like it, Chapter 10 in the free Pro Git book (<https://git-scm.com/book/en/v2>) contains all you need to know.

Back to *OSTree*. It's very similar to Git, but not identical. Just as Git, *OSTree* has a notion of a repository, which is just a directory to store objects. There's one system-level

»



OSTree use-case: Flatpack

If you've been around Linux for some time, you'll remember universal package managers such as *Autopackage*. There were quite a few of them, but they all failed. It's hard to give a single reason why this happened but perhaps the world didn't need another package format to maintain.

What makes Flatpack different is it's not a package manager. It's closer to Docker in that it brings you nearly self-contained, ready-to run applications. This is a clear departure from the previous state of the things where you need to

install all the dependencies (such as a GUI toolkit) before you run an app. The idea of bundling everything with your app is also not new, but Flatpack employs new technologies (such as cgroups, namespaces and OSTree) to make it smart and safe.

When you ship applications as bundles, you trade ease of management for the download size and security. If a bundled library appears vulnerable, you have no single point of update, leaving everything up to the application vendors.

Flatpack tries to balance these requirements via runtimes, which are large shared blocks such as Gnome or KDE platforms applications can build upon. Space-wise, OSTree is a content-addressed filesystem, so each blob is stored only once. Then, communication between the Flatpack app and host operating system is restricted, keeping attack surface to a minimum.

Ubuntu Snap is similar to Flatpack, yet it targets server-side, not desktops. And it has nothing to do with OSTree.

» repository residing in **/ostree/repo**. For any other location, you'll need to provide a **--repo** switch to the tools, or set an **\$OSTREE_REPO** environment variable. Then, *OSTree* uses SHA256 hashes, these are safer (as collisions were found in SHA1 back in 2014) and longer. Meanwhile, *OSTree* doesn't support abbreviations (such as **5fe1c78** for **5fe1c78faf2b430ab937db1cfaf9f3e16592aca3**) so you always have to type hashes in full. There are also branches, but no merges. Where Git sports a tricky revisions mini-language (see **git-rev-parse(1)**), *OSTree* only understands carets (^) that refer to the previous commit.

Object types in *OSTree* are also similar to Git. Both have commits and content (blob) objects. What is called a tree object in Git is split between dirtree and dirmeta in *OSTree*. Dirtree stores filename to hash mapping while dirmeta contains associated metadata, such as UID and GID. The reason is, being Git for operating system binaries, *OSTree* needs to store more metadata. It does it separately for efficiency reasons: if many files share the same extended attributes list (which is often the case), it won't be duplicated.

Storing blobs

Repository format is also a bit different. In Git, blobs are stored compressed. In *OSTree*, it depends. For so-called "bare" repositories, files are stored uncompressed, and *OSTree* "checkouts" them via hard links. "Archive" repositories store compressed files and static deltas which is useful to serve OS images over HTTP. There are a few other storage formats as well, but they are variations of the above two for the most part.

And of course, *OSTree* provides its own tool, dubbed *ostree* that – you guessed it – manages these repositories.

This is where the fun begins, and what we're going to discuss next.

Before we start, make sure you have *OSTree* installed. *OSTree* is a GNOME/Red Hat-backed project, so those on Fedora or CentOS probably win here. If it's not installed, don't worry: *OSTree* should still be in your distribution's repositories. For Ubuntu, you'll need 16.10 and above. For the latest and freshest you can compile it from sources. Yet we encountered dependencies issues on this route, which are certainly solvable but a bit of pain, so better leave this as a back-up option.

Assuming you have the tool installed, this is how you create your first *OSTree* repo to play with:

```
$ ostree --repo=/path/to/repo --mode=bare init
```

```
$ ls /path/to/repo
```

```
config extensions objects refs state tmp
```

Here we initialise a bare repository, which is the default.

Note that in real-world cases *OSTree* typically runs as root, but as we don't operate at OS filesystem tree level, a normal user would do for now.

Repository is initially empty. Let's create some files and then commit them:

```
$ mkdir folder
```

```
$ echo 'Hello, World' > file
```

```
$ ostree commit --branch=playground --repo=/path/to/repo
```

There are a few things to note here, compared to Git. First, the branch is required, because there is no "master" equivalent in *OSTree*. Typically, branches carry path-like names, say **gnome-continuous/buildmaster/x86_64-runtime**, but let's keep things simple here. In contrast, the commit's subject and body (short message and long message, in Git's parlance) are optional by default, and it's fine to store an empty directory. In addition, note that there's no intermediate **ostree add** stage required.

Now change something in the directory:

```
$ echo 'Bye, World' > file
```

and commit this once again to the same branch. This time, add some descriptive subject. Now you can see the log of your changes with:

```
$ ostree log --repo=/path/to/repo playground
```

The output looks much like in Git. Note that ref is again required, because there's no master. It's possible to check any version of this tiny tree you like:

```
$ ostree checkout --repo=/path/to/repo <hash from ostree log>
```

Just remember you need all hash bytes. Note that the tree isn't switched as in Git but rather checked out in a separate directory. In real-world deployments, bootloader integration and systemd tricks are employed to check out the tree you want during the boot.

```
valesini@valesini-ubuntu:~/playground$ ostree --repo=/tmp/repo --mode=bare init
valesini@valesini-ubuntu:~/playground$ ls /tmp/repo
config extensions objects refs state tmp
valesini@valesini-ubuntu:~/playground$ mkdir folder
valesini@valesini-ubuntu:~/playground$ echo 'Hello, World' > file
valesini@valesini-ubuntu:~/playground$ ostree commit --branch=playground --repo=/tmp/repo
9b5ab2d593f3ae1df236537151f10d87fb49a76e4486b33448646924d95f1d
valesini@valesini-ubuntu:~/playground$ echo 'Bye, World' > file
valesini@valesini-ubuntu:~/playground$ ostree commit --branch=playground --repo=/tmp/repo --subject='Some de
841c1a40f2372abcea5ed6c2a5f22407f22b8d54d8a78530e9d0e0b4aed503c2
valesini@valesini-ubuntu:~/playground$ ostree log --repo=/tmp/repo playground
commit 841c1a40f2372abcea5ed6c2a5f22407f22b8d54d8a78530e9d0e0b4aed503c2
Date: 2017-12-25 10:44:16 +0000

    Some descriptive subject

commit 9b5ab2d593f3ae1df236537151f10d87fb49a76e4486b33448646924d95f1d
Date: 2017-12-25 10:44:01 +0000
(no subject)
valesini@valesini-ubuntu:~/playground$ ostree checkout --repo=/tmp/repo 841c1a40f2372abcea5ed6c2a5f22407f22b
valesini@valesini-ubuntu:~/playground$ ls -l
total 12
drwxrwxr-x 3 valesini valesini 4096 янв 1 1970 841c1a40f2372abcea5ed6c2a5f22407f22b8d54d8a78530e9d0e0b4aed
-rw-rw-r-- 1 valesini valesini 11 дек 25 15:44 file
drwxrwxr-x 2 valesini valesini 4096 янв 25 15:43 folder
valesini@valesini-ubuntu:~/playground$ ls -l 841c1a40f2372abcea5ed6c2a5f22407f22b8d54d8a78530e9d0e0b4aed503c
total 8
-rw-rw-r-- 2 valesini valesini 11 янв 1 1970 file
drwxrwxr-x 2 valesini valesini 4096 янв 1 1970 folder
valesini@valesini-ubuntu:~/playground$
```

» Note how *OSTree* resets the timestamps on files it check outs from the repo.

Nix: a different beast altogether

Nix (<https://nixos.org/nix>) doesn't use *OSTree* in any way. It's a distinctive package manager for Unix. Yet it shares a few design goals and is somewhat similar to *OSTree* in spirit. So it makes sense to have a look at *Nix* here to compare both approaches.

Nix manifests itself as a "pure functional package manager". If you don't follow trends in programming languages, this means that building packages don't have side effects, and packages themselves are immutable and deterministic. *Nix* also hashes build

dependencies graph and provides packages isolation through this. So you may have two versions of KDE installed side-by-side. However, *Nix* isn't a content-addressable filesystem.

Nix supports both source and binary deployments. In other words, it can compile everything including the compiler for you, or use cached binary packages. Every package is installed under **/nix/store**, and using the cryptographic hash as a part of the name means you can upgrade and roll back atomically, as in *OSTree*. However, this works at

the package level whereas *OSTree* operates complete filesystem trees.

Officially, *Nix* supports Linux and Mac OS X. There's also – some would say – a niche Linux distribution, NixOS, which, you guessed it, uses *Nix* as an official package manager. The tool also provides a convenient means to manage build environments. If you ever used Docker for this purposes, we suggest you have a closer look. Last but not least, if this all sounds interesting, drop us a line so we can cover *Nix* in a future Administeria article.

Checking out into the current directory is also supported: just add `.` as the last command line argument. It doesn't work the same way as in Git, however. In fact, if you try this with our example setup, *OSTree* would complain:

```
$ ostree checkout --repo=/path/to/repo <hash from ostree log> .
```

error: File exists

This is because we already have both “file” and “folder” in the current directory, and *OSTree* is all about immutable read-only trees. There's a union feature (`--union`) that asks *OSTree* to stack filesystems one on top of another. It keeps all files and directories which *OSTree* considers unchanged and overwrites everything else with the commit's contents. It's mainly used for layering trees much like Docker does with containers. Doing this banishes the above error message.

Remember though, that *OSTree* wasn't meant for mutable filesystem trees. If you check out anything from a bare repository, you really get a hard link – think the second name for blocks on the disk. If you change anything under this name, you effectively modify the repo directly and overwrite your history. This is not what you want, and to prevent this *OSTree* integration scripts create a read-only bind mount for *OSTree*-managed directories in real-world deployment setups. This is also why `/etc` and `/var` – the two typical locations for mutable files in Linux – are effectively out of the *OSTree*'s control.

It's also possible to commit a tree from the tar archive. You just supply `--tree=tar=something.tar` to the “ostree commit” command. This comes handy when you integrate *OSTree* with build systems. Let's have a quick look at how you do this.

A real thing

OSTree has found its way into many projects. Flatpack (<https://flatpak.org>) uses it to distribute and manage both applications and runtimes and touts content deduplication and rollback as driving features. *OSTree* also comes as an update mechanism in Endless OS (<https://endlessos.com>) and forms the foundation for (www.projectatomic.io) Project Atomic. A de-facto standard build/continuous delivery system integration for *OSTree* is perhaps Gnome Continuous (<https://build.gnome.org>).

This being said, Gnome Continuous is experimental, and it seemed to be broken as of a time of the writing. At least, all latest builds were marked as failed. We also had some issues

with the image from a year ago that was available for download. However, integration scripts are still with us, and we can see what they do, and how a typical *OSTree*-managed system is organised.

The build system part is quite straightforward, *OSTree*-wise. It takes sources from Git, compiles them and then commits the result into the *OSTree* repo. An important thing to note here is that Git, not *OSTree*, is deemed to be the ultimate source in this scheme. In *OSTree*, there's a common assumption that files you store in a repo can be regenerated if necessary. There's also a metadata mechanism you can use to store additional info to aid this regeneration, such as Git commit hash or tag.

Typically, you keep around a few latest *OSTree* commits and prune everything else. Gnome Continuous doesn't seem to use this feature, but it's worth considering for your own deployments. The **ostree-prune(1)** man page has all the details. With `ostree prune`, you can delete (garbage collect) unreachable objects, everything older than a threshold, or just specific commits.

With required binaries in *OSTree* repo, build scripts start constructing a so-called deployment. The latter is just an *OSTree* checkout and behaves much like a *chroot*. A family of commands anchored at `ostree admin` are used for that. First, `ostree admin os-init` is called to prepare a new deployment. Then `ostree pull-local` pumps data from the build system repo. Finally, `ostree admin deploy` checkouts the target ref as the new default deployment to become effective after the reboot.

Gnome Continuous ships as a virtual machine image (qcow2) and the filesystem layout inside is also typical for *OSTree*. `/bin`, `/lib` and alike are symlinks to their `/usr` counterparts, and the latter is mounted read-only. Similarly, `/home` and `/root` are symlinks to `/var` so all *OSTree* deployments share them. There's also a `/sysroot` directory, which points to the real filesystem root. This is mainly to give an access to the *OSTree* system repo so you can work with it.

Hopefully, this gives you an understanding of what *OSTree* is all about. It is not often that you have to interface with it, yet it may come as a silent workhorse in a larger system. Actually, the project's name is now libostree to emphasise the fact you'd want to integrate it into your Python or something similar. But if you need an old good CLI, don't panic – it's still present and functional. **LXF**

The screenshot shows the GNOME-Continuous web interface. On the left, a sidebar lists several builds from 20171225.1 to 20171225.7. The main content area displays details for 'Build 20171225.7'. A diagram shows the build process flow: resolve → build/bdiff → builddisks/zdisks → smoketest (with sub-items: smoketest-classic, smoketest-wayland, smoketest-timed) → applicationtest (with sub-items: integrationtest, memusage). Below the diagram, status messages indicate: 'Resolve completed in 165s', 'Changes: mutter: 9 commits [Show]', 'Build completed in 597s', and 'Image building failed'.

» Gnome Continuous was meant a way to try the latest Gnome, but as of now, it feels out of order.

LEDE: Free your routers

Tired of proprietary firmware, **John Lane** gets to grips with LEDE, and discovers that his router is actually quite good...



Our expert

John Lane builds networks for fun and profit. Without LEDE it would be a lot less fun...

Quick tip

Dropbear SSH will use the OpenSSH agent, so you can `git push` without needing a private key on the router. You may need to use `ssh -A` when logging in.

The Linux Embedded Development Environment, or LEDE, is a fork of the long-standing OpenWRT project. This is a Linux distribution for embedded devices, in particular network routers. It's a complete replacement for the vendor-supplied firmware of a wide range of wireless routers. If your device is supported, then installing LEDE on it will release its full capabilities and enable you to fix vulnerabilities when vendors won't.

In this tutorial we'll set up a new LEDE router with basic internet connectivity and perform some customisation as the initial steps towards enjoying the full capabilities of this open source firmware. By doing this we'll look inside this custom Linux distro and learn some of its configuration.

Support me!

Visit the LEDE website at <https://lede-project/toh> to find out if your device is supported and, if so, follow the instructions there to install LEDE on it. The process is very device-specific, as is the level of support that depends on the device's specification – such as its chipset and amount of memory. You might even need to open its case to access its serial console port and may also need to use a soldering iron to connect to it.

Bear in mind that installing custom firmware on your router is most likely to void any warranty it might have, especially if hot molten metal is involved!

We'll assume you already have a LEDE-capable device and have followed the device-specific instructions to install it. We used a BT Home Hub 5 (refer to the Introducing Our Router

box for more information, *above right*) and then installed LEDE 17.01.04, which doesn't suffer from the recently highlighted Krack WiFi vulnerability.

First contact

Before we begin, a little network setup may be required. You can perform preliminary configuration of your LEDE router alongside and without disturbing any other router you may have; you can keep your internet connection working and switch over when ready.

The default LEDE configuration sets the router's IP address to 192.168.1.1 and runs a DHCP server that leases addresses in that **v24** subnet. These are the same defaults that many routers use, so if this clashes with your network, you might want to disconnect your computer from it first. If your network differs (perhaps you've used the **10/8** or **172.16/12** private address ranges), you can use `ip` to manually assign an ad-hoc IP address like this:

```
$ ip add address 192.168.1.10 dev eth0
```

Use an ethernet cable to connect the LEDE router. If your existing router uses the same address then you'll need to temporarily connect the LEDE router directly to your computer to change its address as we describe below. Otherwise, connect it to a spare port on your existing router. Power it on, allow it some time to boot and then check that it's accessible: point a web browser at <http://192.168.1.1> to reach the LuCI (the Lua Configuration Interface) console.

LuCI initially has no password (we'll fix that later). For now, navigate to Network>Interfaces and press the Edit button for the LAN interface. The screen presents options that you should modify to match your own network: choose an address and network mask that's compatible with your local network. Enter the address of your existing router as the gateway and DNS server, then press the Save and Apply button at the bottom of the screen. If you directly connected the LEDE router, you should now reconnect it via your existing router.

Dropbear here

The initial network configuration should enable you to log in to the LEDE console. LEDE uses the lightweight Dropbear SSH implementation designed for embedded systems while being broadly compatible with the OpenSSH used by desktop Linux distributions. Use `ssh` to connect to the `root` account. There's no initial password, so set one up now:

```
computer$ ssh root@LEDE
root@LEDE's password:
root@LEDE:~# passwd
```

```
Don't stop to stomp ants when the elephants are stampeding.

[John@desktop]$ ssh root@gateway

BusyBox v1.25.1 () built-in shell (ash)

  LE  DE
  DE  LE
  LE  DE
  DE  LE

lede-project.org

Reboot (17.01.4, r3560-79f57e422d)

root@gateway:~#
```

Here's what the command line looks like over SSH – with great power comes great responsibility!

Introducing our router...

The router used for this project is a BT Home Hub 5 Type A, a standard issue for BT (A UK ISP) Broadband. There is also a Type B, but its chipset is unsupported. Manufactured by Sagemcom, this router is also issued to some PlusNet customers as the Hub One – the only difference being PlusNet's casing is white whereas BT's is black.

The specification is rather good with dual-band Wi-Fi supporting b/g/n/a/ac standards, a five-port switch, an ADSL2+/VDSL capable modem and a USB port for an external storage device.

If you don't have a supported router, this one is readily available for around £10 on eBay. Bear in mind that you'll need to crack open the case and solder wires on to two very tiny contacts to access the serial port. To help you on your way there's a dedicated community forum over at <http://bit.ly/homehub5>, which is worth checking out.



➤ This tutorial uses BT's router.

Quick tip

To regenerate SSL keys use `rm /etc/uhttpd.*` and restart `uhttpd`.

Changing password for root

New password:

```
root@LEDE:~# exit
```

The password that's just been set may be used to access LuCI, but it's good practice to use key-based authentication for SSH. First upload your public key:

```
computer$ cat ~/.ssh/id_rsa.pub | ssh root@LEDE '(umask 77 && cat >> /etc/dropbear/authorized_keys)'
```

```
root@LEDE:~#
```

```
root@LEDE:~# passwd
```

Changing password for root

New password:

```
root@LEDE:~# exit
```

That uploads your default key, placing it into Dropbear's **authorized_keys** file and ensures that its permissions are correctly set. Now log in again; this time no password should be necessary.

You may now further secure access by disabling password-based SSH logins. This requires modifying LEDE's Dropbear configuration.

LEDE uses a homogenised configuration format called UCI – its Unified Configuration Interface – for its services. This uses text files stored in one place in the filesystem, at **/etc/config**. Being text, you can use `git` to track your configuration changes which, should you wish to do so, is explained in the box (*overleaf*).

Dropbear's configuration is in the UCI **/etc/config/dropbear** file. Edit the options within to look like this (*vi* is available; you can install *nano* if you prefer):

```
config dropbear
```

```
option PasswordAuth off
```

```
option RootPasswordAuth off
```

This illustrates the UCI format. It's plain text broken down into sections with their content indented by whitespace (spaces or tabs). Option values may contain whitespace if they are enclosed within quotation marks. An alternative to manually editing is to use the `uci` tool. To achieve the above result you would enter commands like this:

```
$ uci set dropbear.@dropbear[0].PasswordAuth=off
```

```
$ uci set dropbear.@dropbear[0].RootPasswordAuth=off
```

```
$ uci commit
```

Read https://lede-project.org/docs/user-guide/introduction_to_lede_configuration to learn more about UCI configuration and the `uci` command.

After changing configuration the affected service must be reloaded or restarted. To restart Dropbear:

```
root@LEDE:~# /etc/init.d/dropbear restart
```

You should also enable SSL on the LuCI web server because the root password is used to log in to it and would otherwise cross the network unprotected. LEDE provides a package that does this. You use an `opkg` command to install packages from LEDE's package repository, but its temporary package index must be updated prior to use. Do that and then install *luci-ssl*:

```
root@LEDE:~# opkg update
```

```
root@LEDE:~# opkg install luci-ssl
```

Bear in mind that it uses self-signed certificates, which browsers will warn about. If you wish, you can edit the self-signed certificate parameters in **/etc/config/uhttpd**:

```
config cert 'defaults'
```

```
option days '730'
```

```
option bits '2048'
```

```
option country 'ZZ'
```

```
option state 'Somewhere'
```

```
option location 'Unknown'
```

```
option commonname 'LEDE'
```

The self-signed certificate will be generated when you restart the webserver, after which LuCI must be accessed using https, because http will no longer work.

```
root@LEDE:~# /etc/init.d/uhttpd restart
```

On the air

To replace the existing router, LEDE must become the main internet gateway and Wi-Fi access point for your LAN.

Wi-Fi is initially disabled. The parameters in **/etc/config/wireless** define a hardware device (*wifi-device*) for each radio (a dual band router will have two: one for 2.4GHz and another for 5GHz) and a logical *wifi-iface* interface for each of those. You can edit these parameters to select a channel and to set the SSID identifier and encryption parameters:

```
config wifi-device 'radio0'
```

```
option type 'mac80211'
```

»

» We ♥ hardware hacking! Subscribe at <http://bit.ly/LinuxFormat>

```
» option channel '36'
option hwmode '11a'
option htmode 'VHT80'

config wifi-iface 'default_radio0'
option device 'radio0'
option network 'lan'
option mode 'ap'
option ssid 'LEDE'
option encryption 'psk2'
option key 'my secret key'
```

The initial configuration has a disabled setting. Be sure to remove or comment this out and then initialise the Wi-Fi:

```
root@LEDE:~# wifi up
```

You should now be able to connect a wireless device.

On the wire

The internet, or WAN, configuration is in **/etc/config/network**. The required parameters will depend on both your ISP and how the service is delivered; a basic ADSL2+ service in the UK would be similar to this:

```
config dsl 'dsl'
option annex 'a'
option xfer_mode 'atm'
option line_mode 'adsl'
config interface 'wan'

option proto 'pppoe'
option username 'username@isp.example.uk'
option password 'my password'
option vpi '0'
option vci '38'
option encaps 'vc'
option ipv6 '0'
```

Quick tip

The firmware may provide a **/etc/board.json** file, which describes the default port number map.

```
config device 'wan_dev'
option name 'pppoe-wan'
option macaddr '90:72:82:88:99:00'
```

They will differ for fibre (VDSL), which typically uses **pppoe**. You'll need to experiment to discover what works with your provider.

You should now remove your original router's gateway and DNS from the LAN section. Now is also a good time to change the hostname from LEDE if you'd prefer something else. Look into **/etc/config/system** where you can also set your time zone:

```
config system
option hostname 'gateway'
option timezone 'Europe/London'
```

That completes the basic setup. Turn off the LEDE router and reposition it in your network with your DSL line connected to it and then power up. You should see WAN activity and have internet connectivity through the LEDE router.

A host with the most

At this point you have a basic internet gateway pretty similar to that provided by your ISP but with one major difference: decent firmware. You have a managed switch, a powerful *iptables* firewall, and a DHCP and DNS server. Anything else is probably available through the *opkg* package manager.

You can configure not only DHCP, but static hosts too, each defined in its own "domain" section within the **/etc/config/dhcp** file. **host** blocks reserve DHCP addresses for specific hosts so they always get the same address. Both assign DNS names:

```
config domain
option name myhost
option ip 192.168.10.1

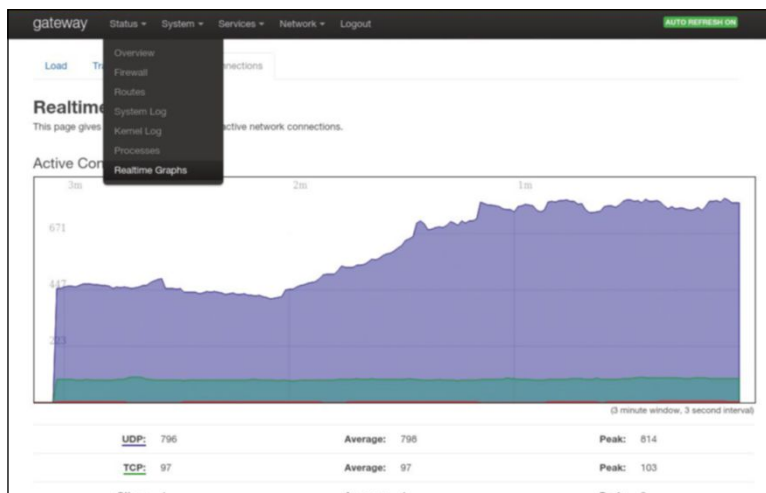
config host
option name otherhost
option ip 192.168.10.2
option mac DE:AD:BE:EF:CA:FE
```

You can also configure alternate DNS names using

```
cname blocks:
config cname
option cname othername
option target myhost
```

Finally, a simple firewall rule makes a host accessible from the internet. Firewall rules are defined in **/etc/config/firewall** and an appropriate port forwarding rule looks like this:

```
config redirect
option name 'Web Server'
option src wan
option dest lan
option dest_ip 192.168.1.2
option dest_port 22
option proto tcp'
```



» LuCI's real-time graphs help you keep on top of how your network is being used.

Version control

You can use Git to track changes to your UCI configuration files. If you like, first use LEDE's **opkg** package manager to install **git**:

```
root@LEDE:~# opkg update
root@LEDE:~# opkg install git
```

Then initialise a new repository in **/etc/config**, which is the UCI configuration directory,

and then make an initial commit as follows:

```
root@LEDE:~# cd /etc/config
root@LEDE:~# git init
root@LEDE:~# git add .
root@LEDE:~# git commit -m 'Initial commit'
```

From here onwards, you can commit your configuration changes as you wish. At this point

you could set up a remote repository and **git push** the config to it; you could customise Git to your liking and perhaps add a **.gitignore**.

One option that you might find useful disables Git's colour output; the Busybox terminal that LEDE uses doesn't support colour:

```
root@LEDE:~# git config --global color.ui false
```


Firewall rules work on zones, so the above redirects from the wan zone to the lan zone. Elsewhere in the same firewall configuration file you'll find definitions that specify what network interfaces are in each zone. The interfaces are defined in the network file. After editing it, remember to reload the relevant configs:

```
root@LEDE:~# /etc/init.d/dnsmasq restart
root@LEDE:~# /etc/init.d/firewall restart
```

Get orf my LAN...

Many networks are being infiltrated by devices other than computers. Phones and tablets, children's toys, so-called personal assistants like Alexa and whatever else the increasingly popular internet of things may bring. You can keep such devices out of your cosseted LAN using a de-militarised zone, or DMZ, by creating a virtual network interface. We'll set up a DMZ to help learn more about LEDE's networking configuration.

If you look at the network configuration, you'll see that the existing LAN interface is bound to **eth0.1**, where the suffix tells us it's a virtual interface. We can create a new virtual interface: copy the lan configuration, rename it to **dmz** and then use another suffix:

```
config interface 'dmz'
option type 'bridge'
ifname 'eth0.2'
proto 'static'
ipaddr '172.20.0.1'
netmask '255.255.0.0'
config device 'lan_dev'
option name 'eth0.2'
option macaddr '54:64:d9:01:02:03'
```

We also allocate the new interface a MAC address and its own IP subnet (**ipaddr** and **netmask**). To have the DHCP server issue addresses requires an addition to the **/etc/**

config/dhcp file:

```
config dhcp 'dmz'
option interface 'dmz'
option start '256'
option limit '255'
option leasetime '12h'
```

The start value defines the first address in the block. In this example that would be 172.20.1.0 and the limit is the number of addresses in the block.

The firewall needs to allow the DMZ to access the internet. We allocate a DMZ firewall zone and specify a forwarding rule:

```
config zone
option name dmz
list network dmz
option input ACCEPT
option output ACCEPT
option forward ACCEPT
```

```
config forwarding
option src dmz
option dest wan
```

You can allocate one or more of the router's Ethernet ports to the DMZ. The ports attach to a managed Ethernet switch within the router that's configured using switch blocks in the network file. Our default looked like this:

```
config switch
option name 'switch0'
option reset '1'
option enable_vlan '1'
```

A second access point

You can configure LEDE as a simple wireless access point. This is a good way to extend the range of your network. There are a number of ways to do this – see <https://wiki.openwrt.org/doc/recipes/bridgedap> – but we connected a pair of routers together using Ethernet.

Because the second router is only a wireless access point and network bridge, it doesn't need the DSL capability. Only the wireless and Ethernet components are used.

We connected an Ethernet cable between ports configured on each router as trunks so that it carried both lan and dmz traffic. We also disabled and removed the DHCP and firewall from the access point because those functions are provided by the main gateway:

```
$ opkg remove --force-depends luci-app-firewall firewall dnsmasq
```

The network and wireless on the access point is set up in a similar way to the gateway.

```
config switch_vlan
option device 'switch0'
option vlan '1'
option ports '0 1 2 4 5 6'
```

Here you can see the default VLAN for the LAN zone (the **vlan** parameter reflects the network interface suffix that we described earlier). It specifies the ports it includes. A point to note here is that these port numbers may bear no relationship to the order of the physical ports on the router. Some numbers may not be used (such as port 3 on our device) and there's an additional logical port to which the router's ethernet interface (for example, **eth0**) is internally connected. This is port 6 on our router, which has five physical ports.

Pack your trunk

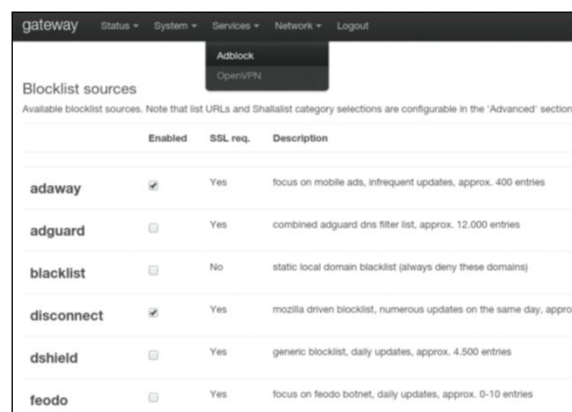
The other thing to understand here is that a port may carry traffic for multiple virtual interfaces. These are known as Trunk ports and are specified by using a "t" suffix, such as in port 6 in the example. Network traffic between trunk ports is "Tagged" with the VLAN identifier which, as such ports can be in multiple VLANs, enables the switch to direct traffic to the appropriate one.

To allocate switch ports to the DMZ all we need to do is remove them from the LAN block and then add them to a new DMZ block:

```
config switch_vlan
option device 'switch0'
option vlan '1'
option ports '0 1 5 6t'
```

```
config switch_vlan
option device 'switch0'
```

»



» You can select block lists with the adblock LuCI extension.

```
» option vlan '2'
option ports '2 4 6'

You can also create separate Wi-Fi access to the DMZ,
which may be useful to provide a guest access. Augment the
"wireless" config file:

config wifi-iface 'dmz_radio0'
option device 'radio0'
option network 'dmz'
option mode 'ap'
option ssid 'LEDE Guest'
option encryption 'psk2'
option key 'guest secret key'
option isolate 1
```

This looks similar to our main SSID, but connects to the dmz network instead of the lan. The optional isolate prevents Wi-Fi guests from seeing each other.

Remember to restart the appropriate services: network, wireless, firewall and dnsmasq (for DHCP). Or just reboot!

Protect ads, block kids and more!

Other things that you can do include parental controls (<https://lede-project.org/docs/user-guide/parental-controls>), setting up an OpenVPN server (<https://lede-project.org/docs/user-guide/openvpn.server>), and one of the more simple and effective additions, which is running an ad-blocker.

It's relatively simple to install the LEDE adblock package and it starts working automatically. The optional LUCI package adds a Services>Adblock menu where you can configure the service. You can choose which blocklists are used: there are several listed of which the default configuration uses three:

```
$ opkg install adblock luci-app-adblock
```

Another interesting application is traffic monitoring, useful whether your broadband connection has a capped bandwidth or just because you want to keep watchful eye over the network. The easiest to setup is the Netlink Bandwidth Monitor, *nwblmon*, which adds some nice pages to LuCI. Install it with `opkg install luci-app-nlwmn` and select the new LuCI "Bandwith Monitor" page.

Go with the NetFlow

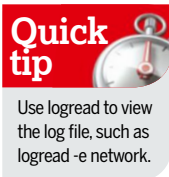
An alternative, or additional, method you can use is NetFlow. This protocol originated on Cisco routers as a way to collect data about network traffic. Devices send NetFlow data to a collector running on another host. One open source collector is nfcapd, part of NFDump (www.nfdump.sourceforge.net) and can be partnered with a web application called *Netflow Sensor*, or *NfSen* (www.nfsen.sourceforge.net) to make network traffic data available in a web browser.

To send NetFlow data, LEDE supports *softflowd*, which is easily installed (`opkg install softflowd`) and configured in `/etc/config/softflowd`:

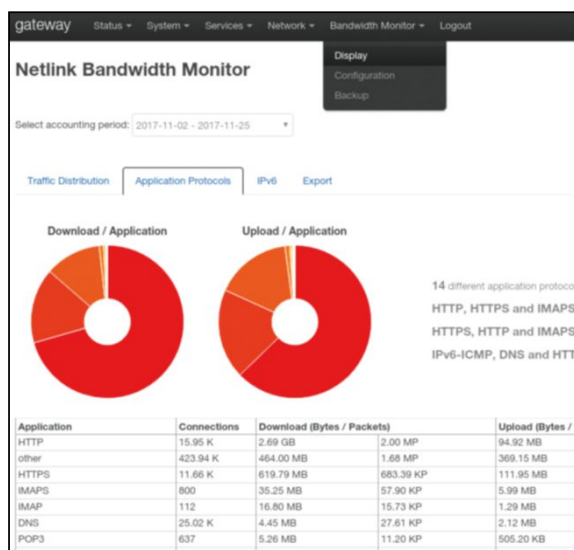
```
config softflowd
option enabled '1'
option interface 'pppoe-wan'
option host_port '192.168.1.10:2055'
option export_version '9'
option track_ipv6 '1'
```

In this example, `host_port` is the IP address where NetFlow packets should be sent. This is another Linux box on your network and so you'll need to install *nfsen* and its dependencies on it.

LEDE is not just for routers. It also has uses on other non-network devices. For the full list of supported devices see the official Table of Hardware at <https://lede-project/toh>. Not that it also won't be LEDE forever because an agreement has been reached to re-merge with the OpenWRT project and this will involve returning to the original name. Until that happens, the LEDE fork remains the most active and better supported one – just bear in mind that we expect that brand to disappear at some point. **LXF**



Use logread to view the log file, such as `logread -e network`.



» **Monitoring bandwidth is easy with LEDE.**

Beating buffer bloat

Even though it's pretty much de rigueur to complain about your internet service provider (either on social media or at dinner parties), your internet connection may not be as bad as you thought.

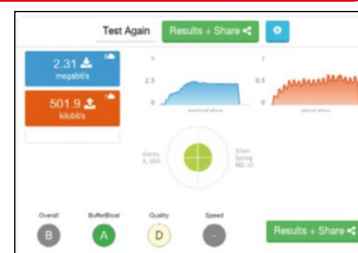
If you suffer buffering on your network then it might be suffering from buffer bloat. This is caused by network devices buffering too much data when the network is under load. Put simply, a user performing a big download can impact another user's video phone call. The problem is explained in a little more detail at www.bufferbloat.net.

LEDE has a *Smart Queue Management* package to help address the problem of buffer bloat. You can install it and also remove the redundant QoS packages, if installed:

```
$ opkg remove qos-scripts luci-app-qos
$ opkg install luci-app-sqm
```

Configure in `/etc/config/sqm` or navigate to Network>SQM QoS in LuCI.

You can then test your network for buffer bloat with the DSLReports speed test at www.dslreports.com/speedtest for example.



» **SQM may help you get the most out of a mediocre broadband connection.**

» **We're #1 for Linux!** Subscribe and save at <http://bit.ly/LinuxFormat>



The best new open source software on the planet

LXFHotPicks



Alexander Tolstoy
is missing snow in St Petersburg, but he's got some great open source programs to play with instead!

LXFHotPicks

Darktable » Ddgr » Midnight Commander » Flameshot » Liquidshell » Posterizer » Andriod File Transfer » Cating » Desktopfolder » Bemuse » HexGL

Photo editor

Darktable

Version: 2.4 **Web:** www.darktable.org

Over a year has passed since the last stable update of *Darktable* (see **LXF235** for an indepth look) was released, so we were pleased to review this outstanding member of the open source software world.

Darktable is a professional-grade tool for processing RAW images taken with a D-SLR or any other type of camera that's capable for shooting images in the RAW format. Outside the Linux ecosystem, *Darktable* competes with such commercial heavyweights as Adobe's *Lightroom* and Apple's *Aperture*, and this alone confirms that Linux, in general, is mature enough to

meet the tough requirements of professional photographers.

Rendering RAW images and carrying out tasks such as the live preview of effects takes up a lot of CPU power. Thankfully, *Darktable* makes use of sophisticated, optimised code to deliver a fine degree of performance. As an example, we chose to view a very large bitmap image of the night-time view of



» **Darktable hides great power behind the mask of humble image browser.**

“Darktable is a tool for processing RAW images taken with a D-SLR”

the Earth from NASA. This is a massive JPEG file that weighs in at 220MB. Most Linux image viewers failed to open it, even on systems fitted with 8GB of RAM, but *Darktable* does the job swiftly even on low-end configurations.

So what's new in version 2.4 of *Darktable*, apart from almost 3,000 commits and over 340 closed issues? The latest release features the new Haze Removal tool, which is handy for improving shots that have been taken outdoors, and fixing the effects of a dirty lens. The Local Contrast module now features a local laplacian mode, and the Undo history now supports masks. Fujifilm compressed RAFs and floating point HDR DNG files are fully supported too.

There are many more tasty new additions for you to discover, and if you regularly play around with RAW photos, you'll definitely notice improvements under the hood, too. Features such as grouping steps in the Undo history, the Tone Curve tool, de-mosaicing filter, map location tool and others all have been improved for a better and smoother user experience.

Get the latest *Darktable* copy from the project's website or check your distro's repositories for the update.

Shedding light on the Darktable interface

Light and dark

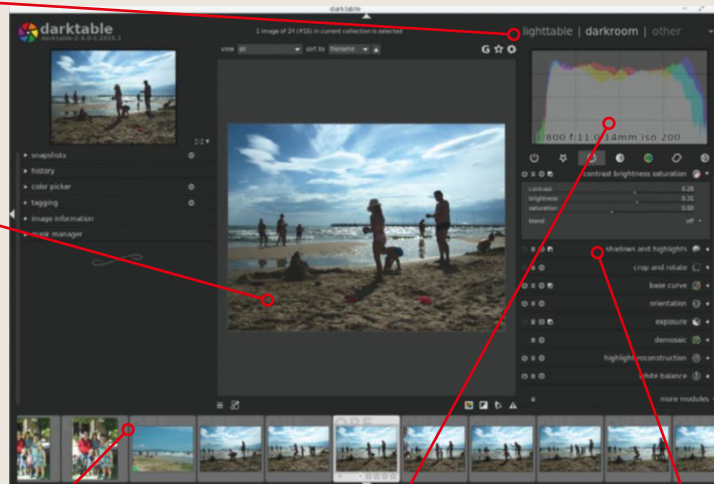
Lightroom is where you browse and sort images. Darkroom is for editing the ones you picked.

Live preview

Darktable sports a fast rendering engine that applies all your changes virtually on the fly.

More at your fingertips

Keep an eye on other images without leaving *Darktable*.



Colour precision

Review the colour histogram and camera settings that were used to take the image.

Fix and enhance

Use various colour tools to fix the brightness, shadows or contrast of your image.

Search tool

Ddgr

Version: 1.2 **Web:** <https://github.com/jarun/ddgr>

Linux gives you an unprecedented level of computing freedom, whether you want to get rid of the command line entirely, or do most of your work in a terminal window. We're here with some news for command line ninjas, who'll probably be happy after discovering *Ddgr*.

It's a DuckDuckGo search helper from the author of *Googler* – the same tool that carries out Google queries from the command line. We believe that the privacy-aware DuckDuckGo is a better candidate for *HotPicks*, so let's move on with it.

The idea is for a command line internet search tool that can functionally supersede a text-based web browsers (such as *Lynx*) when it makes sense to do so. Navigating a web site with your keyboard could be tricky, but at the same time you wouldn't want to fire up a web browser just to make a simple search in Google or DuckDuckGo.

Ddgr (where 'r' supposedly means 'results') is a simple, Python-powered utility that doesn't want you to read documentation pages in order to get started. Just type your query after the *ddgr* command and you quickly generate the first ten results.

Supply the number of the result and hit Enter to instruct *Ddgr* to launch your default web browser with the appropriate link active. It couldn't be simpler, yet *Ddgr* has some lovely advanced extras for you to explore.

First, it cleans away all visual clutter and ads, so you get only the useful text entries with readable formatting and sensible colouring. *Ddgr* supports HTTPS proxy, Bash-style text

“Ddgr is a pleasure to use, because it uses very little screen space”



➤ Sometimes it's enough to simply review the search results – no browser needed.

completion, keywords and more. You can also edit the number of search results per page, hide the user agent or change the default URL handler.

Ddgr is a pleasure to use, because it uses very little screen space to display your search results, enabling you to keep your terminal window small.

There are many pre-built *Ddgr* packages available from the official web site, but in case you're not in the list, just run `sudo make install` and get ready for the DuckDuckGo terminal fun!

Terminal file manager

Midnight Commander

Version: 4.8.20 **Web:** www.midnight-commander.org

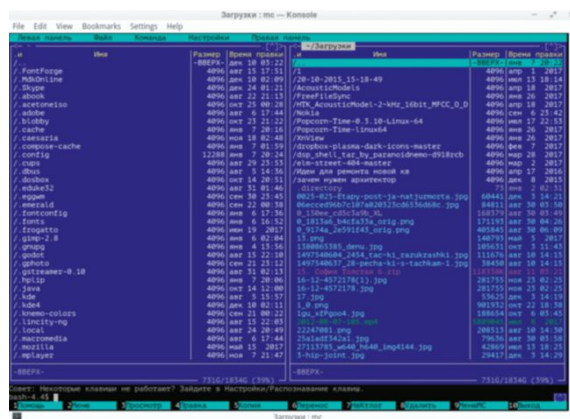
We're glad to see the latest update to this mature and venerable file manager, which is popular among both older UNIX folks and the younger generation. In our experience most people will know how to install *Midnight Commander* (or *MC*) on Linux, because it's included in almost all distributions by default. It's slightly less common than GNU C Library, but far more widespread than *LibreOffice*, for instance.

Midnight Commander offers a tried-and-tested twin-panel approach to managing your files and directories. Interestingly, the traditional deep-blue panels with aquamarine selections is the type of working environment where you would probably use the 'directory' term instead of the Window-centric 'folder'. But we digress.

MC is an application that, despite its age, is still capable of being used on a

daily basis. In real life, *MC* doesn't feel as outdated as say, *Emacs*, so you can start using it without further ado. A bar running along the bottom of the screen features keyboard tips for basic file operations, such as F3 for viewing, F4 for editing, F5 for copying files and so on. However, there are still some tricks you should know, to make your *MC* session a bit more productive.

The first one prevents you from exiting *MC* if you feel like it's not your cup of tea: press Ctrl+O to switch to the command line without leaving the program (press it again to get back). Use Shift++ to select files and directories using your own wildcard, or



➤ Browse your files like your grandpa did. *MC* is as solid as a cast-iron pan.

“Midnight Commander is included in almost all distributions by default”

press Shift+* to select everything in current view. *MC* also enables you to open a directory in a new tab: press Esc+O to open the contents on another panel. Hit Ctrl+U to swap panels, or go with Alt+I to make both panels identical.

Exploring *Midnight Commander* is a good reason to read its documentation. There are dozens more keystrokes in this file manager that can save your time and boost your productivity.

Screenshot tool

Flameshot

Version: 0.5.0 **Web:** <http://bit.ly/flameshotot>

The days when the Linux desktop offered a selection of basic screenshot tools are definitely long gone. Nowadays, when asked what computing platform has now the richest and the most complete set of screen grabbing software, we'll just smile and point at the chubby, lovable Tux. To further support our findings, here's another rival to *Ksnip* (LXF223) and *Screencloud* (LXF226), both of which are previous holders of the *HotPicks* mantle.

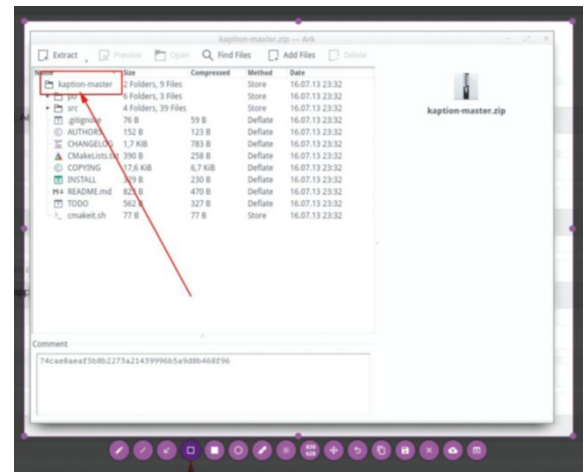
Flameshot is a great tool for capturing custom areas of a desktop, and in this sense it's more a fully fledged snipping tool than just a replacement for your PrintScreen button. The application has some distinctive features that are worth pointing out. First, *Flameshot* creates as many as three items in your menu: one for configuring the app, another for starting it in a silent mode in the

System tray and one more launcher for firing up the grabber directly.

Flameshot suggests that you may want to change the default selection of annotation tools, set the file naming template using common variables (date, time and so on) and customise the colours of the snipping tool itself.

The range of annotating tools is very good, although not perfect. *Flameshot* can't place auto-numbered labels, but it still enables you to draw lines, arrows, ellipses and rectangles. A blur tool means you can hide sensitive information on your grabs, and there's an Imgur upload plugin, too. We liked the speed and ease of use of this screenshot-sharing feature. Although

“It's more than just a replacement for your PrintScreen button”



➤ The frame and those circular buttons are present even after you've drawn something.

Imgur is the only cloud option in *Flameshot*; it does the job well enough when it comes to sharing images publicly. Alternatively, you can always save your grabs as a local file in any file format (provided by Qt5 image plugins), or open it in another application.

As a parting shot, *Flameshot* ships with a spectacular set of translations and an experimental Wayland support for both Plasma 5 and Gnome desktops.

Desktop shell

Liquidshell

Version: GIT **Web:** <https://github.com/KDE/liquidshell>

We believe whatever graphical library an application is based on, it ought to run smoothly in any desktop environment you choose. Modern desktops, such as Gnome, KDE Plasma, Mate and Cinnamon, are all able to make GTK- and Qt-based apps look native.

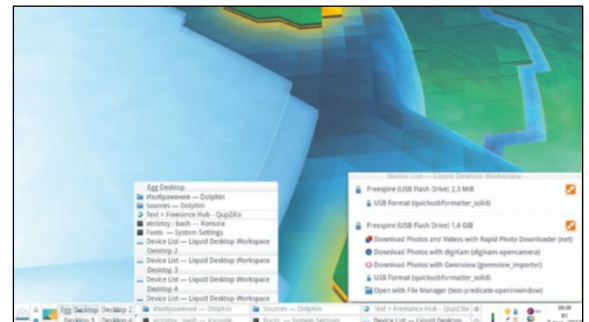
However, this particular *HotPick* is something more desktop-specific, because it tries to supersede the 'plasmashell' process with a more lightweight alternative. *Liquidshell* is an alternative desktop shell that's designed to work with the typical KDE desktop (it's officially hosted by the KDE project, as you may have guessed from the Github page address).

Plasmashell isn't particularly resource-hungry, yet *Liquidshell* is even sleeker in operation, mainly because it has a simpler design and doesn't use Qt's declarative QtQuick language.

Liquidshell is a simple shell built from classic QtWidgets controls, including the bottom panel that features a System tray, clocks, the launcher menu, the set of virtual desktops and other convenient elements. It starts almost instantly and feels very easy to use.

Although you'll need a collection of various KDE Frameworks 5 development parts to build *Liquidshell* from source, the main obstacle is how to run it. This time you don't have a separate 'session' entry in your greeter. Instead, you'll need to add *Liquidshell* to your auto-start list and then shadow-ban the normal Plasmashell process. To

“Liquidshell is a simple shell built from classic QtWidgets controls”



➤ If Plasma desktop still crashes for you, ditch it and use Liquidshell instead!

do so, place the file with the following two lines as another auto-start item:

```
[Desktop Entry]
```

```
Hidden=true
```

The next time you log in, *Liquidshell* should be running on top of Plasmashell, and you can even kill it entirely (**killall plasmashell**). The new shell has links to your default applications, and also a System tray that lists devices and active programs.

We had a great time with *Liquidshell*. The whole experience was like a more solid KDE session, or even a feature-packed LxQt release from the future!

Image optimizer

Posterizer

Version: GIT **Web:** <http://bit.ly/lxf-posterizer>

We've previously written quite a lot about different encoders, optimisers and the tricks you can do to reduce the size of your image library without compromising quality. It looks like we're now short of new lossless tools, but still able to blow your mind with some yet-very-good apps for lossy compression.

Lately we've come across *ImageAlpha*, a MacOS-only graphical program that can slim down your PNG files footprint, especially if they have an alpha channel (transparency). That application relied on a command-line tool known as *MediantCut-Posterizer*, or *Posterizer* for short. As the author writes, it has two modes: a lossy averaging filter (blurizer) that removes noise from the image; and optimal posterization using Median Cut quantization to reduce the number of unique colours in the image with minimal visual distortion. We ran our

own tests and discovered how this tool affected our test true-colour PNGs with transparency. The most common usage probably looked like this:

```
$ posterizer -Q 75 in.png out.png
```

The program reduced the size of our test image by an impressive 30 per cent without introducing any visible artefacts. Playing with the quality ratio (`-Q`) revealed that even with very low figures *Posterizer* still produced decent results, and even though the image no longer looked smooth, it was still less than half its original file size. The *Posterizer* could be very easily turned into a 'blurizer' by appending the `-b` parameter and, again, playing with quality ratio.

“The program reduced the image's size by an impressive 30 per cent”



› One of these images was slimmed down by 50 per cent using *Posterizer*. It's probably the one on the right.

Lossy averaging didn't reduce the file size instantly (the output file was larger than the input), but the idea is that averaging alone just prepares the way for further compression. You can re-export a blurred PNG image in *Gimp* and see how much more the file size can be reduced.

There are many ways you can benefit from using *Posterizer*, besides optimising web sites. For example, Raspberry Pi and many low-power IoT devices will work better with smaller PNG icons and other PNG content.

File transfer tool

Android File Transfer

Version: GIT **Web:** <http://bit.ly/android-file-transfer>

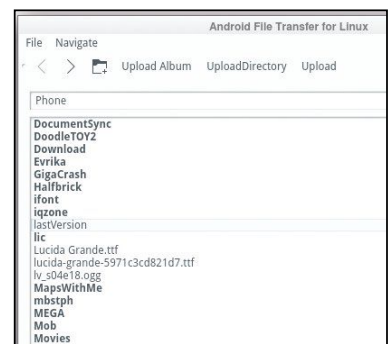
Most modern smart phones are equipped with decent cameras, which is the reason of the shrinking market of standalone compact cameras. Even if many of us upload our smartphone-taken shots to various cloud storage services, there's still a need to download images to a computer, from time to time.

Android-based phones no longer use the classic USB mass storage connection type in favour of MTP (media transfer protocol), and while the latter is supported in all major Linux desktops, there are still stability problems with MTP in Linux (at least in Dolphin). We were looking at ways to transfer the DCIM directory from an Android device to our Linux box until we came across *Android File Transfer*. This is a great standalone GUI tool that solves all the problems related to MTP support in Linux.

Android File Transfer will only launch if you've already connected your phone with your Linux computer via USB cable. If not, you'll only see a warning notification that the program couldn't find any MTP device.

The main window of the tool is quite simple, and perhaps for some people it's too simple for a file manager. Bear in mind, then, that *Android File Transfer* is only a file copying tool rather than a fully featured file manager. It enables you to browse your device's file structure and select which content you wish to download. You can also upload some files onto your phone, or delete existing files and directories there.

“It solves all the problems related to MTP support in Linux”



› Easily transfer files from your Android smartphone at full speed.

So, in many regards *Android File Transfer* mimics the feature set of *Rapid Photo Downloader* with some important differences. There's no preview thumbnails for images and videos, but you are able to manipulate files on your smartphone.

From our perspective, the main advantage of *Android File Transfer* is its very reliable support for MTP. It may sound obvious, but in real life things like GMTP, GVFS and MTFs are not bug-free, so that you can experience random freezes or corrupted files. Using *Android File Transfer* solves the problem gracefully!

HotGames Entertainment apps

Music game

Bemuse

Version: v38 **Web:** <http://bit.ly/lxf-bemuse>

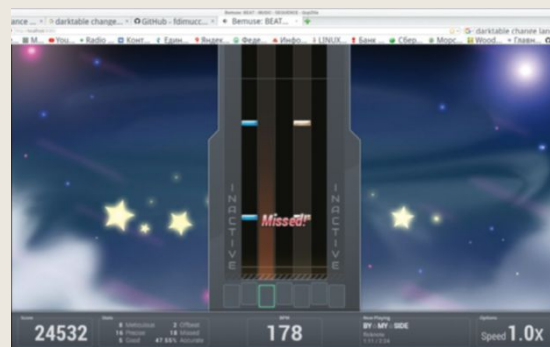
Bemuse combines a vertical-scrolling space shooter with a rhythm-based music game. You have to press the notes whenever they appear on screen. You can use various external devices, such as MIDI keyboards and turntables, but a regular PC keyboard is fine, too. It helps if you can play a piano, but even if you can't, *Bemuse* can help you gain some strength in your fingers and wrists.

In order to perform well (and score more points) you need to boost your reaction times and get used to a custom key layout. *Bemuse* starts with a tutorial for novice players, where you learn how and when to use the seven keys (S, D, F, Space, J, K and L) and also enjoy the suggested Electronicore and Trans

dance tracks. Depending on your results, *Bemuse* awards you with grades, from F (not too good) to A (ninja!) and invites you to promote yourself in the online Bemuse ranking list. This feature is only available when you play Bemuse online, because the local installation won't let you upload your scores.

Bemuse ships with a decent selection of default songs, and you can plug in custom songs. The game accepts BMS files, which you can explore at <http://bmssearch.net> (note that the site's in Japanese). We

“Bemuse comprises a collection of modern web technologies”



► A fun game that combines music and reaction times.

spent a few hours trying to advance to the next grade with a couple of default *Bemuse* dance tracks and didn't get bored at all!

Bemuse comprises a collection of modern web technologies, such as React and HTML5, and is very easy to roll out at your localhost. You'll need to have *Nodejs* and *Yarn* installed in your system to un just these two commands: `$ yarn` and `$ npm start`.

Don't forget to rest your fingers after long gaming sessions!

Racing game

HexGL

Version: GIT **Web:** <http://bit.ly/hex-gl>

These days it's possible to enjoy high-quality 3D games running inside a web browser window. Both Sourceforge and Github are populated with great HTML5-based games, some of which take new heights in browser gaming.

HexGL is one of those games. This is a futuristic racing simulator with stunning graphics, and a good test-bed for your browser's WebGL capabilities. Although the game itself isn't massive, it does require decent OpenGL performance from your video driver, otherwise you'll be forced to run it on a low detail setting.

In *HexGL* you control a jet aircraft that blazes along a fixed course (even though the jet is flying above the course, it can't stray from it). The game has a very fast-paced single player mode: simply complete all

three laps as fast as you can. The course is set within an urban landscape, with skyscrapers and picturesque aerial views. Initially, your jet has a reliability score of 100, but every time you hit part of the course it becomes damaged. So if you don't fly smoothly, you can destroy your plane before reaching the finish! The course features acceleration spots that boost your speed, but can lead you to suffer damage if you accelerate into a sharp turn.

The menu screen of *HexGL* has some basic settings that you may want to change before going to the race. The most important one is graphics quality:

“HexGL is a futuristic racing simulator with stunning graphics”



► Don't forget to use breaks before hot turns and use accelerators wisely!

it's set to Very High by default, and if your machine isn't up to the job then the game won't launch at all. Press F5 to reload the page and set the quality to High to fix it.

Running the game is simple as long as you don't need to mess with lots of dependencies. *HexGL* relies on the Python's SimpleHTTPServer package, so all you need to do is execute the `$ python -m SimpleHTTPServer` command and open `index.html` in your web browser.

Image viewer

Catimg

Version: 2.4.0 **Web:** <https://github.com/posva/catimg>

We have another gem that will make your daily session at the Bash console more bright and colourful. It's not about taming your `~/bashrc` or playing with pseudo-graphic ASCII effects, but it does affect how images are displayed in a terminal.

Catimg is a simple tool that will save you a little time when you need to view an image's contents, but don't want to fire up a dedicated image viewer. The name essentially plays on the phrase 'a cat' and the common UNIX `$ cat` command. So without further ado, let's see how it goes:

```
$ catimg /path/to/file
```

Catimg uses *ImageMagick* to convert images, and as such can handle a variety of image formats. Pictures are rendered right inside your terminal, and they don't look pseudo-graphic at all, because *Catimg* renders every pixel as a coloured square symbol. The quality

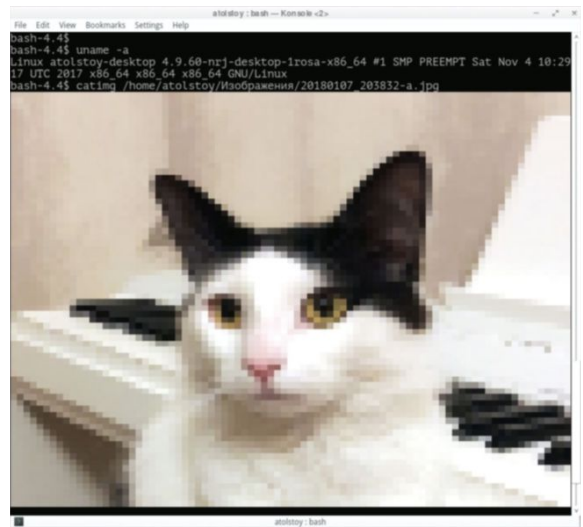
depends on the physical size of the image, and you should be aware of this. Large pictures look fantastically detailed, although they seem to be overly zoomed in.

You can open an image at a smaller size by using the `-w` parameter:

```
$ catimg -w 200 /path/to/file
```

The above command assumes that the width of your terminal is 200 pixels and if your image is wider then it'll be downscaled. It'll be also rendered at a smaller resolution, because the size of terminal's pseudo-pixels is fixed.

Catimg works best when used in graphical mode. If you try it without either X11 or Wayland then it'll print a very rough version of your picture, with



➤ Using *Catimg* gives you another reason to stay in terminal for just a bit longer.

recognisable outlines but the wrong colours on show. The full-resolution graphical mode is recommended because it provides *Catimg* with an adequate palette. On the other hand, viewing small graphic files, such as icons, avatars, thumbnails is perfectly suited to *Catimg*. You can see all the details and still keep using your terminal session.

“Catimg renders every pixel as a coloured square symbol”

Desktop accessory

Desktopfolder

Version: 1.0.5 **Web:** <http://bit.ly/desktopfolder>

One of this month's *HotPicks* was *Liquidshell*, which is KDE-specific, so we decided to feature another similar program.

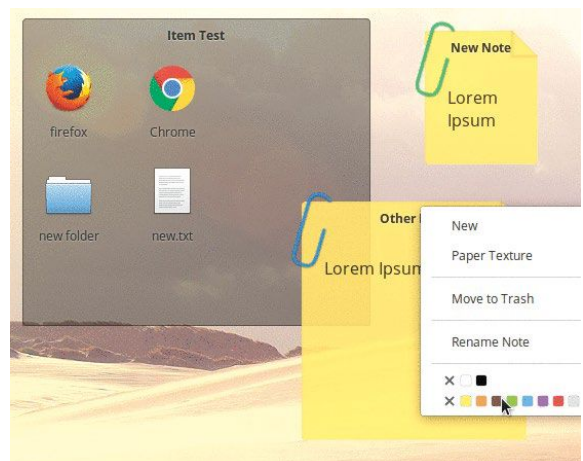
Desktopfolder is specific to ElementaryOS, or, to be more precise, to its Pantheon desktop. Purists can argue that Pantheon is also available for Arch and Fedora, but still the majority of users prefer to run it within its 'home' eOS.

Despite the fact that one of our test machines runs ElementaryOS and we use it frequently, it's often hard to review stylish fancy apps from Elementary's brand new *AppCenter*, simply because they are too 'Elementary' and don't provide enough features for a decent review.

Luckily, this isn't the case for *Desktopfolder*, a cool addition to Pantheon that brings its desktop back to life. As you might know, the Pantheon

desktop is intentionally empty, without any means to put files or icons on it. If you're not happy with this state of affairs, *Desktopfolder* is the solution. This application creates floating panels that serve as containers. You can put various things inside it, including files, folders, launchers, images and more.

Desktopfolder provides enough settings to customise panels for different needs, so in the end you can have multiple panels of different types: a file container, a sticky note or an image frame. It works much like a basic set of plasmoids, only in this case it's bound to the ElementaryOS tool set and written using Vala.

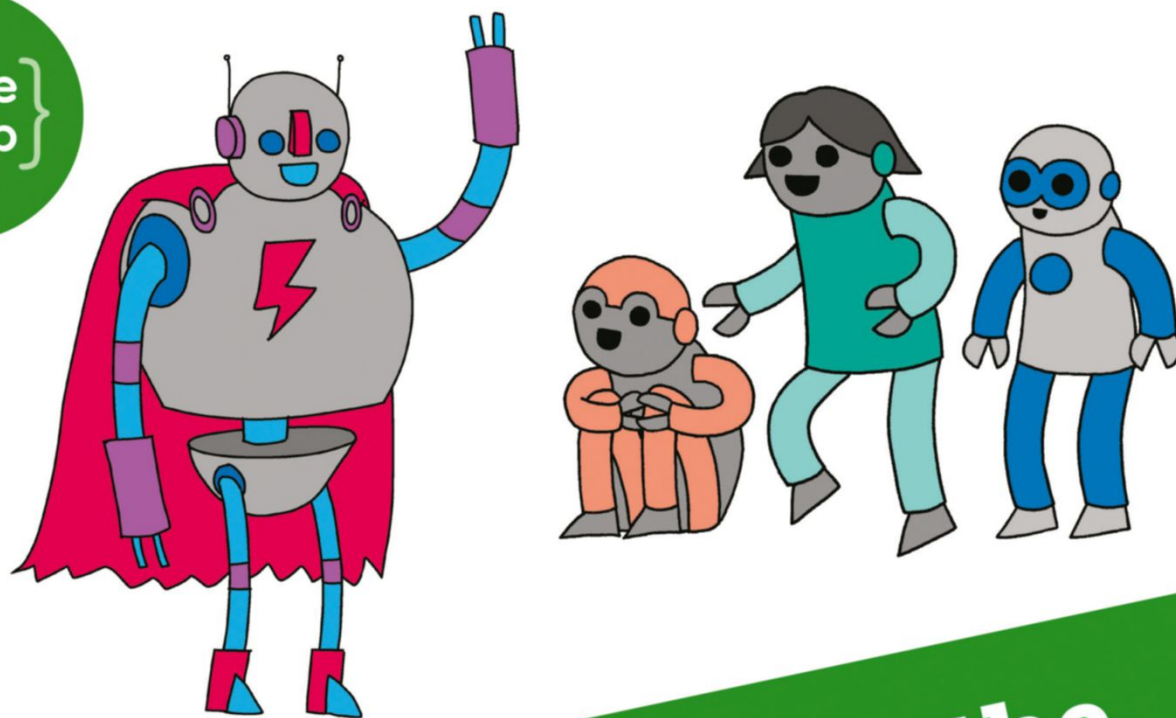
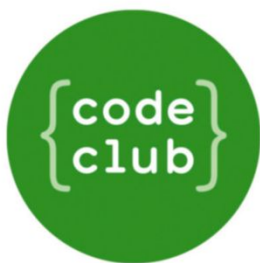


➤ Material-styled widgets sitting on top of the ElementaryOS desktop? Sounds promising...

We really enjoyed messing with *Desktopfolder* panels, especially resizing, adding colour tags and dragging some content onto a panel.

The concept goes against ElementaryOS' ideals, but this is Linux and we can still exercise our freedom of choice. *Desktopfolder* is an open source project that provides pre-built DEB packages. You can get it installed in just a few clicks via *AppCenter*. If you're using Pantheon, don't miss it!

“A cool addition to Pantheon that brings its desktop back to life!”



**Can you help inspire the
next generation of coders?**



Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!
So to find out more, join us at **www.codeclub.org.uk**

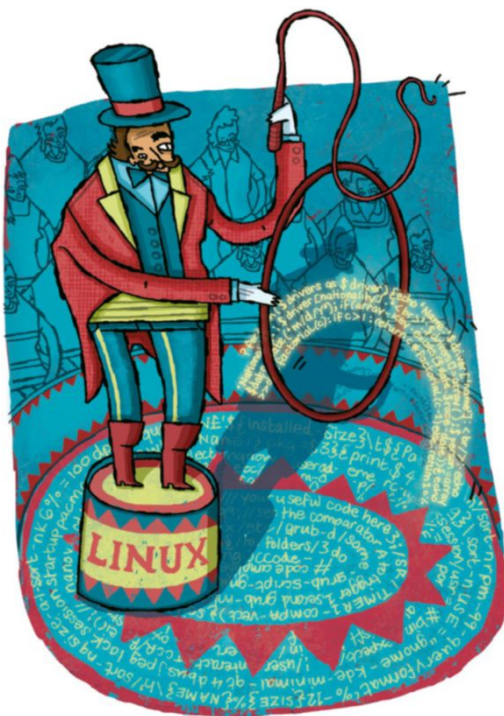
CMake: Get the best build tools

Mihalis Tsoukalos teaches you about the CMake project, its configuration files and how to visualise project dependencies using Graphviz.



Our expert

Mihalis Tsoukalos is a UNIX administrator, a programmer, a DBA and a mathematician who enjoys writing technical articles. He's also the author of Go Systems Programming.



```
3. mtsouk@mail: ~/docs/article/working/CMake.LXF234/code/hw/build (ssh)
hw$ ll
total 8
-rw-r--r-- 1 mtsouk mtsouk 78 Dec 30 21:45 MakeLists.txt
-rw-r--r-- 1 mtsouk mtsouk 94 Dec 30 19:47 hw.c
hw$ mkdir build
hw$ cd build/
hw$ cmake ..
-- The C compiler identification is GNU 4.9.2
-- The CXX compiler identification is GNU 4.9.2
-- Check for working C compiler: /usr/bin/cc -- works
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting C compiler ABI info
-- Detecting CXX compiler ABI info
-- Configuring done
-- Generating done
-- Build files have been written to: /home/mtsouk/docs/article/working/CMake.LXF234/code/hw/build
hw$ make
Scanning dependencies of target LXF
[100%] Building C object CMakeFiles/LXF.dir/hw.c.o
[100%] Linking C executable LXF
[100%] Built target LXF
hw$ ll
total 36
```

➤ The output that's generated by the **cmake** command when dealing with a simple **CMakeCache.txt** configuration file for a project that has just one source file written in C.

CMake suite maintained and supported by Kitware (kitware.com/cmake).

Should you wish to look at the numerous options supported by **cmake**, you can execute the **cmake --help** command. After this brief introduction, we're now ready to start working and building projects with **CMake**.

MakeBasic

The simplest project you can have is one with a single source file. So, imagine that you have a C file called **hw.c**, which contains the C code of the famous *Hello World* program, and that you want to manage it using **CMake**.

The name of the **CMake** file that holds the configuration of that naive project will be **CMakeLists.txt**, which is the default configuration file for all **CMake** projects. The contents of that simplistic and naive **CMakeLists.txt** file are the following:

```
cmake_minimum_required(VERSION 3.0)
PROJECT( LXF )
add_executable(LXF hw.c)
```

The first line defines the minimum **CMake** version required for this project, the second line specifies the name of the project and the third line specifies the files involved (**hw.c**) in the creation of the executable file as well as the filename of the executable file (**LXF**).

Please note that it's considered good practice to have a directory named **build** inside the root directory of your project in order to have a cleaner root directory. Additionally, because **cmake** doesn't provide a command for cleaning up,

A cornerstone of the build process is **CMake**, an open source project largely supported by Kitware. If you want to build and compile code like a pro then you've got to get in the know! Here, we're going to show you how to use **CMake** for compiling, linking and installing software on Linux and UNIX machines. We'll then use the **Graphviz** output from **CMake** to visualise the dependencies of a **CMake** project. Remember that if you have a properly setup **CMake** configuration file, **cmake** will do most of the boring work for you. However, it should be noted that you should use the GNU **make** utility to process a **CMake** project.

You'll most likely install **CMake** using your usual package manager, which will make the whole process much simpler for you. You might also want to install some extra packages to make your life easier in the long run – on an Ubuntu Linux, you can do that by executing the following command:

```
$ sudo apt-get install cmake cmake-data cmake-doc cmake-extras
```

Then, you can find out which version of **CMake** you have by running this command:

```
$ cmake --version
cmake version 3.10.1
```

Quick tip

Learn more about CMake at <https://cmake.org> and <https://cmake.org/documentation>. Visit https://cmake.org/Wiki/CMake/Testing_With_CTest for details on CTest, and read up on CPack at https://cmake.org/Wiki/CMake/Package_With_CPack.

What can CMake do for you?

First of all, you should know that *CMake* stands for Cross platform Make, which means that it can create build files for all kinds of UNIX machines as well as Windows machines with the help of *Microsoft Visual Studio*.

CMake is ideal for dealing with big and complex projects, and can recognise which

compilers to use for the source files that it has to process on its own. Indeed, this auto-detection feature is a key reason for using *CMake*. There's also a curses version of *cmake* called *ccmake* that enables you to work more interactively with *cmake*. Using *ccmake* while learning *cmake* is good practice. Nevertheless, when you become

more experienced with *CMake*, you'll prefer the command line version of it because it's a quicker way to work with *CMake* projects.

Bear in mind that although *CMake* can achieve many things, you're not obliged to either make use of all the capabilities of *CMake* at once, or apply it to every one of your projects!

you'll just need to delete the **build** directory when you want to clean up your project.

After having a syntactically correct *CMake* file, you'll need to execute the **cmake** command with a single parameter, which will be the path to the root directory of your project. This will automatically generate some additional files including a **Makefile** that will be used by the *make* utility. So, the required steps are as follows:

```
$ cd hw
$ ls -l
total 12
drwxr-xr-x 2 mtsouk mtsouk 4096 Dec 30 21:45 build
-rw-r--r-- 1 mtsouk mtsouk 78 Dec 30 21:45 CMakeLists.txt
-rw-r--r-- 1 mtsouk mtsouk 94 Dec 30 19:47 hw.c
$ mkdir build
$ cd build
$ cmake ..
```

The last command will create lots of output that can be seen in the screenshot (*left*). Executing the **make** command afterwards will create an executable file called **LXF** in the root directory of your *CMake* project. If you don't want to manually create the **build** directory you can execute **cmake -H. -Bbuild** from the root directory of the project.

Finally, if *cmake* can't find a **CMakeLists.txt** file in the specified directory, you'll see the following error message:

```
$ cmake /tmp
CMake Error: The source directory "/tmp" appear to contain
CMakeLists.txt.
Specify --help for usage, or press the help button on the
CMake GUI.
```

As the *CMake* file shown here is relatively straightforward, we won't deal with it any further. However, if you're familiar with Makefiles then it would be very interesting to have a quick look at the generated Makefile.

The next section will present the *CMake* configuration of a more realistic project with multiple source files written in C++.

Starting C++

This C++ project will have three source files that are needed for creating the final product, which will be an executable file. To avoid overcomplicating matters, the C++ code of this project will be fairly simplistic. A *CMake* project can also create a shared library or a static library – the general idea behind processing such projects with *cmake* is the same as in creating an executable file.

The steps and the commands for creating the structure of a new *CMake* project called 'simple' are the following:

```
$ mkdir simple
$ cd simple
$ touch CMakeLists.txt
```

```
$ mkdir build
$ mkdir myLibrary
$ mkdir myLibrary/include
$ mkdir myApplication
$ touch myApplication/main.cpp
$ touch myLibrary/include/aClass.h
$ touch myLibrary/aClass.cpp
```

The first thing that you see in this project is that although there are multiple directories, there's only one **CMakeLists.txt** file. Additionally, apart from the central C++ file that's named **main.cpp** and contains the **main()** function of the project, there are two additional files: one C++ source file (**aClass.cpp**) and one header file (**aClass.h**). In these two files you'll find the implementation of a C++ class that's used in **main.cpp**.

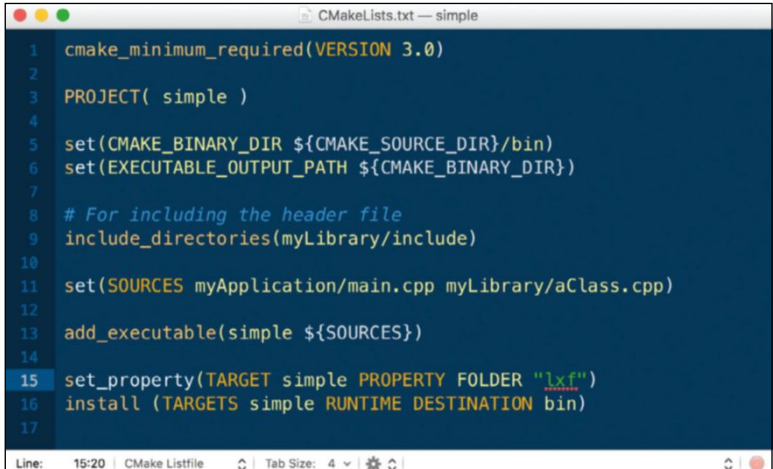
The contents of the main **CMakeLists.txt**, which can be found in the root directory of the project, can be seen in the screenshot (*below*). Note that you could have used multiple **CMakeLists.txt** files instead, but for such a relatively simple project that approach wouldn't be necessary.

Many new and interesting things can be seen in this **CMakeLists.txt** file. First, notice that the **CMAKE_BINARY_DIR** variable in combination with the **EXECUTABLE_OUTPUT_PATH** variable specify the directory where the executable file of the project is going to be placed.

Then, **SOURCES** specifies the two C++ source files of the project. Additionally, **include_directories()** tells CMake where to find **aClass.h**. After that, **add_executable()** specifies that the final product will be named **simple** and that it'll be based on the value of the **SOURCES** variable.



Quick tip
If you want to learn more about *Graphviz* you can read the Using *GraphViz* and the DOT language tutorial in issue 219 of *Linux Format* or visit <http://graphviz.org>, which is the official *Graphviz* site.



```
1 cmake_minimum_required(VERSION 3.0)
2
3 PROJECT( simple )
4
5 set(CMAKE_BINARY_DIR ${CMAKE_SOURCE_DIR}/bin)
6 set(EXECUTABLE_OUTPUT_PATH ${CMAKE_BINARY_DIR})
7
8 # For including the header file
9 include_directories(myLibrary/include)
10
11 set(SOURCES myApplication/main.cpp myLibrary/aClass.cpp)
12
13 add_executable(simple ${SOURCES})
14
15 set_property(TARGET simple PROPERTY FOLDER "LXF")
16 install (TARGETS simple RUNTIME DESTINATION bin)
17
```

» This screenshot shows the *CMake* code of the **CMakeCache.txt** configuration file used for the **simple** project that deals with C++ source files.

» **Love code? Love LXF!** Subscribe now at <http://bit.ly/LinuxFormat>

» The last line of **CMakeLists.txt** tells *CMake* that the generated binary file will be installed on the bin directory of the installation directory.

It's now time to try to use **CMakeLists.txt**. As you already know, you'll first need to execute the **cmake** command before continuing with the make command:

```
$ ls -l
$ cmake -H. -Bbuild
$ cd build
$ make
```

If there's an error in the *CMake* configuration file, you'll see error messages that will look similar to the following:

```
CMake Error at CMakeLists.txt:7 (INSTALL):
INSTALL called with unknown mode DESTINATION
```

The cause of the first error message is an unknown variable in **CMakeLists.txt**, which was corrected in the **CMakeLists.txt** version that you have:

```
CMake Warning (dev) in CMakeLists.txt:
  No cmake_minimum_required command is present.  A line
  of code such as
```

```
cmake_minimum_required(VERSION 3.10)
```

should be added at the top of the file. The version specified may be lower

if you wish to support older *CMake* versions for this project. For more

information run "**cmake --help-policy CMP0000**".

This warning is for project developers. Use **-Wno-dev** to suppress it.

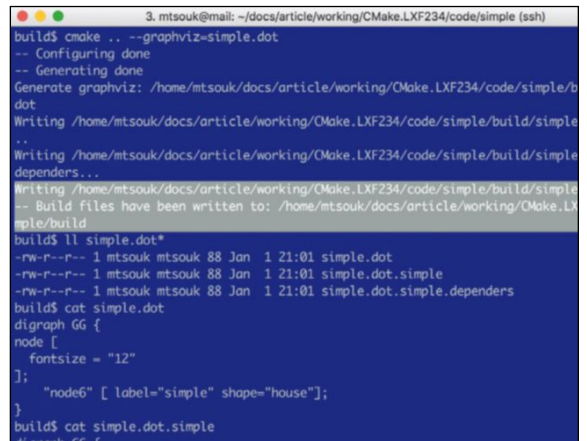
The second message is a warning message telling you that you should define the minimum *cmake* version required for this project. As you can see, both messages are very clear.

The screenshot (*below*) shows the output you'll get when processing the **CMakeLists.txt** file with *cmake*.

CMake variables

The executable file of *CMake* supports a range of variables that can help you change various things on a *CMake* project without the need to alter **CMakeLists.txt**. The most useful variable is **CMAKE_INSTALL_PREFIX**, which helps you define the installation path of your project. The default value of **CMAKE_INSTALL_PREFIX** is **/usr/local** on all UNIX machines. When testing a project, a good candidate for the installation directory would be the **/tmp** directory. The **CMAKE_BUILD_TYPE** option variable specifies the build type that will be built. The

```
build$ make
Scanning dependencies of target simple
[ 50%] Building CXX object CMakeFiles/simple.dir/myApplication/main.cpp.o
[100%] Building CXX object CMakeFiles/simple.dir/myLibrary/aClass.cpp.o
Linking CXX executable ../bin/simple
[100%] Built target simple
build$ cd ..
simple$ ll bin/
total 12
-rwxr-xr-x 1 mtsouk mtsouk 11120 Jan 1 20:45 simple
simple$ tree -d
.
├── bin
├── build
│   ├── CMakeFiles
│   │   ├── 3.0.2
│   │   ├── CompilerIdC
│   │   └── CompilerIdCXX
│   ├── CMakeTmp
│   ├── simple.dir
│   ├── myApplication
│   └── myLibrary
├── myApplication
├── myLibrary
└── include
```



» **Figure 4:** This Figure shows how you can generate Graphviz files that show the dependencies between the targets of a *CMake* project.

valid values for the **CMAKE_BUILD_TYPE** variable are Debug, Release, RelWithDebInfo and MinSizeRel. The Debug value turns on the debug flags on the generated files. However, when creating the final version of a project, you'll need to use the Release value for **CMAKE_BUILD_TYPE**.

Finally, there's **CMAKE_LANG_FLAGS**. You should replace **<LANG>** with the string that's related to the programming language of your choice – this is for defining the compiler flags which will be used. For C, the value of **<LANG>** should be C whereas for C++ the value of **<LANG>** should be CXX. Therefore, for a C++ project, this option variable should be **CMAKE_CXX_FLAGS**.

For the full list of the available option variables you can visit <https://cmake.org/cmake/help/v3.0/manual/cmake-variables.7.html>.

Compile and install

In this section you'll learn how to compile and install the simple *CMake* project, using the **make** command.

So, in order to compile the project, after successfully running *cmake*, you'll need to execute the **make** command from the build directory of the C++ project. Should you wish to compile your project with the debug flags turned on, you should use the **CMAKE_BUILD_TYPE** variable from the command line, as follows:

```
$ rm -rf build
$ make build
$ cd build
$ cmake .. -DCMAKE_BUILD_TYPE=Debug
$ make
```

The use of the **objdump(1)** command will verify that the generated executable file contains debugging information:

```
$ objdump --syms bin/simple | grep debug
0000000000000000 l d .debug_aranges
0000000000000000 .debug_aranges
0000000000000000 l d .debug_info 0000000000000000
.debug_info
0000000000000000 l d .debug_abbrev 0000000000000000
.debug_abbrev
0000000000000000 l d .debug_line 0000000000000000
.debug_line
```

Quick tip

The main advantage of *CMake* over *GNU make* is that *CMake* is cross platform. Additionally, the configuration files of *CMake* would be simpler and shorter than the configuration files of a similar *GNU Make* project. If you're working with a simple project then either of them will do the job, but for more complicated projects *CMake* is more effective.

» This is the output you'll obtain from executing **cmake** on the root directory of the simple project and the **make** command on the build directory of the project.

» **Get your regular Mihalix fix** Subscribe now at <http://bit.ly/LinuxFormat>

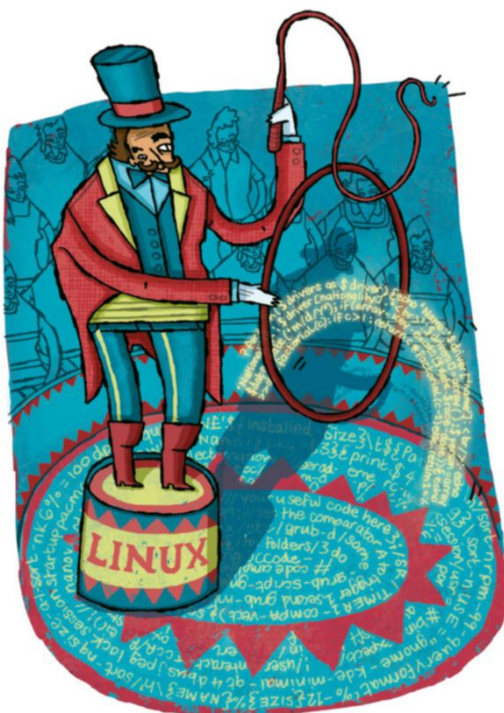
Arrays: a drop of golden sun

Nate Drake guides you through the most fun that you can have with Python arrays while still sober and in full control of your limbs.



Our expert

Nate Drake is a freelance technology journalist. After coding the Magic 8 Ball array, the script now makes all-important life decisions for him. Outlook not so good...



Source: www.kornale.com/nodes.php?lang=en&page=40981

› Arrays can store data related to ray tracing, a rendering technique for generating realistic images by tracing the path of light as pixels. All these objects are computer generated.

designed to only hold one type of object, such as integers. You determine the kind of data the array will hold when you create it. To get started with arrays, first ensure you have *python3* installed on your machine. Most popular Linux distributions bundle this by default.

Get importing

Next, fire up your Terminal or IDLE (see boxout, *above right*) and import the array module with:

```
From array import *
```

Next create the array **myarray**. This will contain only integers (specified by the type code **i**) from 1 to 6.

```
myarray=array('i',[1,2,3,4,5,6])
```

Finally, have Python list the values in the array:

```
For i in myarray:
```

```
print(i)
```

If you're using IDLE, you can use F5 at this stage to run your script, otherwise save your file with a meaningful name use *Terminal* to run `$ sudo python3 <meaningfulfilename>.py`.

You can also print individual values (or elements as they are known) from the array in exactly the same way as for lists, for instance run `print (myarray[0])` to view the first element, which in this case is the number 1.

While you may feel a glow of success at this stage, in its current format this Python script is only useful to those coders who are unable to count past the number five. To this end, create a new file and paste in the code below. This Python script uses the secret module to add some

Those readers who have any experience of grappling with the coils of the Python will already know that the programming language is designed to handle data structures efficiently. The most basic of these is the Python list, which can be written as a sequence of comma separated values. For instance, to create a list of your favourite foods you can declare a list as follows:

```
list1 = ['chocolate', 'bananas', 'caviar', 'jelly beans']
```

Once you've done this, you can easily recall values by specifying the numerical index of an item in the list. Run `print(list1[0])` to display the first item in the list (in this case it's chocolate). The process of searching for and displaying values based on the index is sometimes known as slicing.

Lists are very versatile in that you can store multiple data types such as strings and integers. You've also seen how easy they are to implement. If you only wish to store a list of your weekly groceries in Python, you may need nothing else. However, when it comes to processing larger amounts of data efficiently, Python also incorporates a module that enables you to create arrays.

Much like lists, arrays are a type of data sequence, although the garden variety array module in Python is

Quick tip

Python uses zero-based indexing. In other words the first element in any array has an index of zero, the second is one and so on. If this doesn't make sense ask yourself how many years old you were, the day you were born!

Be an IDLE fellow

Creating a Python script can be as simple as using your favourite text editor to create a file with the extension **.py**, then adding your code. However, this can prove tricky when it comes to indenting lines as well as when you're making minor revisions and need a way to execute your scripts quickly each time.

IDLE is Python's integrated development and learning environment. IDLE includes a multi-window text editor with features above and beyond simple word processing programs such as smart indents, call tips and auto complete of common terms. Once a script has been saved, you can launch it using IDLE's Python shell window by pressing F5. This is ideal

for interactive scripts as the window automatically colourises input, output and error messages.

IDLE is a cross platform application. If you have a Raspberry Pi, you'll find it preinstalled in the Programming section of the Applications menu – choose Python 3 (IDLE) for this tutorial. IDLE is also installed in Ubuntu repositories. Simply open Terminal and run **sudo apt-get update** then **sudo apt-get install idle3**.

The Python website has an extensive user manual for IDLE. Visit <https://docs.python.org/3/library/idle.html> for more information. For the purposes of this tutorial, remember that after you've executed a script you can

continue to interact with it through entering commands via the shell window, for example to append an element to an array.



» IDLE comes preinstalled for both Python 2 and 3 in Raspbian. The program is available in the repositories of most flavours of Linux.

randomness to your slicing, turning it into a rough and ready electronic dice game:

```
import secrets
from array import *
myarray=array('i',[1,2,3,4,5,6])
roll = secrets.choice(range(0,len(myarray)))
print (myarray[roll])
```

The value **roll** here is the minimum value of the **myarray** index (0) and the maximum length which is specified as **len(arrayname)**. You can use this at any time to retrieve the number of elements in a standard Python array.

Manipulating arrays

While so far you've learned how to create and list elements in an array, you can also manipulate them while a program is running. You may wonder why you'd bother doing this when you can just stop the script and edit the code itself. However, this is useful for keeping track of data that you've processed and adding new information. For instance, if you're running a virtual casino (see below) you may want to add new players and winnings, as well as keep a record of cards already played.

To insert a value in a specific position in an array, use **insert**. For instance **myarray.insert(0,1)** will set the very first element in the array to 1. This doesn't delete existing data, because the other items will be shuffled round accordingly.

If you simply want to add new elements to the end of an array, you can use **append**, for instance:

```
myarray.append(7)
```

However, if you want to add a large number of variables, you'd need to do this individually for each of the new values which doesn't make for very tidy code. A more efficient way to do things is to create a new array with all the values you'd like to append, then extend the existing array as follows:

```
newvalues=array('i',[8,9,10])
myarray.extend(newvalues)
```

Call on Numpy

While the standard Python array module is a great way to familiarise yourself with arrays, it's difficult to use it in more advanced scenarios. For instance, there's no quick and easy way to store string values, nor to create a multidimensional

array (more on this later). Fortunately Numpy (Numerical Python) has specifically been designed for scientific computing and contains tools to enable you to create ultra-efficient and powerful arrays. To get started, first make sure you have the *Pip* installer on your machine by opening *Terminal* and running:

```
$ sudo apt-get install python3-pip
```

Next enter the following:

```
$ sudo pip3 install numpy
```

The convention for invoking the Numpy module when writing a script is **import numpy as np**. From there you can declare a Numpy array with **np.array**. One of the most immediate advantages of using Numpy to create your array is that it can automatically detect the type of data you're entering into the array such as a string or integer, enabling you to store both. In keeping with the electronic dice you created earlier, use your favourite editor to create an empty file named **magic8.py** and paste the following:

```
import secrets
import numpy as np
myarray=np.array(['It is certain.','It is decidedly so','Without a
doubt.','Yes definitely.','You may rely on it.','As I see it, yes'
»
```

```
>>>
===== RESTART: /home/lubuntu/magic8.py =====
You may rely on it.
>>> np.insert(myarray,3,'Absolutely!')
array(['It is certain.', 'It is decidedly so', 'Without a doubt.',
'Absolutely!', 'Yes definitely.', 'You may rely on it.',
'As I see it, yes', 'Most likely', 'Outlook good', 'Yes',
'Signs point to yes', 'Reply hazy, try again.', 'Ask again later.',
'Better not tell you now', 'Cannot predict now.',
'Concentrate and ask again.', 'Do not count on it.',
'My reply is no.', 'My sources say no.', 'Outlook not so good.',
'Very doubtful.'],
      dtype='<U26')
>>> np.append(myarray,'No')
array(['It is certain.', 'It is decidedly so', 'Without a doubt.',
'Yes definitely.', 'You may rely on it.', 'As I see it, yes',
'Most likely', 'Outlook good', 'Yes', 'Signs point to yes',
'Reply hazy, try again.', 'Ask again later.',
'Better not tell you now', 'Cannot predict now.',
'Concentrate and ask again.', 'Do not count on it.',
'My reply is no.', 'My sources say no.', 'Outlook not so good.',
'Very doubtful.', 'No'],
      dtype='<U26')
>>> |
```

» Use **numpy.insert** to place an element at a specific position in your array. If you simply want to add a value to the end, use **numpy.append**.



Quick tip
Running the commands discussed here will make your six-sided die 10-sided. You can find a completed example of this script (**1d10.py**) ready for download from <http://bit.ly/10-sided-dice>.

» Don't miss the next issue Subscribe now at <http://bit.ly/LinuxFormat>



```

» , 'Most likely', 'Outlook good', 'Yes', 'Signs point to yes', 'Reply
hazy, try again.', 'Ask again later.', 'Better not tell you
now', 'Cannot predict now.', 'Concentrate and ask again.', 'Do
not count on it.', 'My reply is no.', 'My sources say
no.', 'Outlook not so good.', 'Very doubtful.'])
answer = secrets.choice(range(0, myarray.size))
print (myarray[answer])

```

You can also download this script from <https://github.com/nate-drake/arrayexamples/blob/master/magic8.py>.

This little five line gem creates a numpy array containing text values, namely the 20 possible answers given by users of the classic Magic 8 Ball toy. Notice that you can use `myarray.size` to determine the overall length of a numpy array instead of the more cumbersome `len`.

“Reply hazy, try again”

Aside from using text values and a more efficient type of array, the basic premise of this script is exactly the same as the previous ones you created in that it will print values based upon a random index. The standard Magic 8 ball contains 10 positive answers, five negative and five non-committal. If you want to make it a little fairer by adding another negative answer use the `numpy.append` function for instance:

```
np.append(myarray, 'No')
```

If you do this using IDLE, the program will automatically regurgitate a new list of all the array elements, including the one you just added.

In the case of this script, elements are chosen randomly, so their order doesn't matter. If however you do want to add a value in a specific position to a numpy array, use the `numpy.insert` function, for instance:

```
np.insert(myarray, 3, 'Absolutely!')
```

This will place the word 'Absolutely!' as the fourth element in the list (remember the index counts from zero).

2D or not 2D

While Numpy can be used to create one-dimensional arrays, for example a list of values, it's perfectly capable of creating a multidimensional table of values, too. This isn't a hard concept to grasp; however, most of the online documentation related to Numpy tries to explain this in as convoluted way as possible, which is why we'll devote some time to a more reasonable explanation here.

The simplest type of multidimensional array has two dimensions, which are known as axes. A good usage example would be creating a two-dimensional array to store both X and Y coordinates on a graph and indeed Numpy is designed specifically for storing complex scientific data like this.

» Here the array deck has two axes. Although [2,11] and [3,11] both contain the string Queen, the card suits are distinguishable due to their positions within the array.

```

===== RESTART: /home/lubuntu/drawcard.py =====
Your card is the 2 of Hearts.
>>> print(deck)
[['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']
 ['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']
 ['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']
 ['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']]
>>> print(deck[2,11])
Queen
>>> deck[2,11] = 10
>>> print (deck)
[['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']
 ['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']
 ['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' '10' 'King']
 ['Ace' '2' '3' '4' '5' '6' '7' '8' '9' '10' 'Jack' 'Queen' 'King']]
>>>

```

Create multi-dimensional arrays

If a two-dimensional array can be thought of as a spreadsheet, then a 3D array is akin to a cube. Numpy is perfect for creating these types of N-Dimensional arrays through using ndarray. True to its name there's no theoretical limit on the number of dimensions your array can have but for the time being, imagine that you want to create a virtual blackjack game which uses five decks of cards. If you followed the steps above you may have already created an array with a shape of 4,13. In other words, each card can be located based on suit and number. To get started with creating a 3D array to hold data on whether a deck card has been played, you can

use some of the NumPy Array creation routines. For instance, run:

```
import numpy as np
decks = np.zeros((5,4,13), dtype=np.int)
```

This will create a new array with all elements set to zero. The `dtype` determines the kind of data to be stored in the array. By default, this is float64. If you prefer to add 1 instead, substitute zeroes with ones. You can also create an array filled with values you've specified. See <https://docs.scipy.org/doc/numpy/user/basics.creation.html> for a full list of array creation routines. Once your multidimensional array has been created, you can manipulate it in the same

way to the previous Numpy arrays you've created for example, `decks[1,3,10] = 99`.

```

>>> print(decks)
[[[0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]]
 [[0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]]
 [[0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]]
 [[0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0 0 0 0]]]
>>> decks[1,3,10] = 99
>>>

```

» Use `numpy.zeros` and you, too, can have your very own multidimensional array.

» **Want even more coding?** Grab a bookazine at <http://bit.ly/LXFspecial>

An easier way to understand two-dimensional arrays is as being akin to rows and columns on a spreadsheet. For instance, if you had a deck of cards you could arrange them into an array of four rows (representing clubs, diamonds, hearts and spades) and 13 columns representing the cards for each suit. Each card in this two-dimensional array would have a unique index (or tuple as it's known for multi-dimensional arrays). The overall tuple of a multi-dimensional array is known as its shape. In the example above this would be 4,13.

Start off by creating a new file named **drawcard.py** and paste the following text. This results in a simple script using a two-dimensional array to draw at random from a deck of cards, as follows:

```
import secrets
import numpy as np
# Suits follow the Bridge order: Clubs (0), Diamonds (1),
Hearts (2), Spades (3).
deck=np.array(['Ace',2,3,4,5,6,7,8,9,10,'Jack','Queen','King'],[
'Ace',2,3,4,5,6,7,8,9,10,'Jack','Queen','King'],['Ace',2,3,4,5,6,7,
8,9,10,'Jack','Queen','King'],['Ace',2,3,4,5,6,7,8,9,10,'Jack','Queen',
'King']])
randomsuit=secrets.choice(range(deck.shape[0]))
suit = ['Clubs', 'Diamonds', 'Hearts', 'Spades']
randomcard=secrets.choice(range(deck.shape[1]))
yourcard=deck[randomsuit[randomcard]]
print('Your card is the ' + yourcard + ' of ' + suit[randomsuit]
+ '.')
```

Visualise your array

If you're using IDLE, after running this script, enter the command `print(deck)` to output the two-dimensional numpy array as text. This arranges the 13 cards for each suit in four rows, allowing you to visualise the array. The variables `deck.shape[0]` and `deck.shape[1]` represent the number of elements along each dimension (4 and 13, respectively). These are used to randomly select both a suit such as clubs and then a card within said suit such as the Jack. Note that the Numpy array is happy to store both numerical elements such as 9 and string elements such as King.

The index (or tuple) for the suits is an integer value, so the list suit is used to marry this up to an actual name (clubs, diamonds and so on). This isn't necessary for the array to function, but enables the program to output the suit name in a meaningful way.

Any card in this array deck can be located using the format `deck[X,Y]`. This can be put to good use with the variable `yourcard`, which allocates a chosen card in the deck based on random values. You can also use this to display a specific card – for example, `print(deck[2,11])`. If you want to change values to represent their points value for Blackjack for instance, simply declare the variable, such as `deck[2,11]=10`.

Shattered dimensions

As this tutorial is designed to give you a basic introduction to arrays, there's very little to go wrong. Raspberry Pi users who want to run the sample scripts may encounter some difficulties as the 'secrets' module used to generate random numbers only works with Python 3.6.

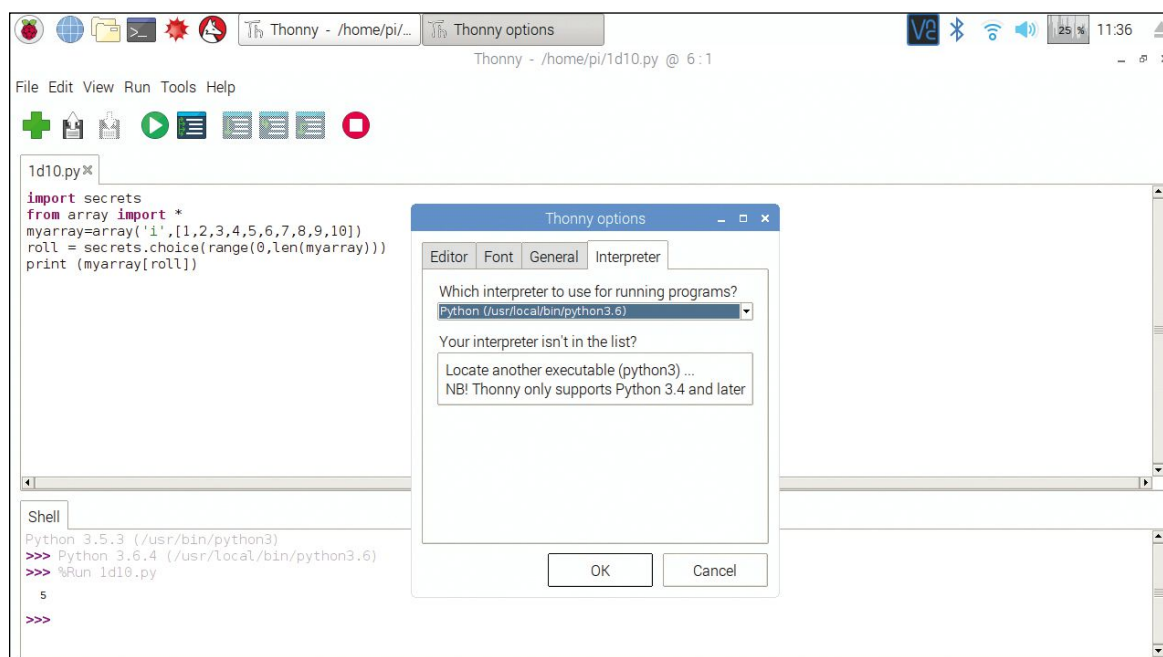
If you're comfortable with compiling software, you can download the latest stable version of Python from www.python.org/ftp/python (currently 3.6.4) and install it via the Pi Terminal. Use the command `$ make altinstall` to keep the existing version of Python. Alternatively, rework the scripts yourself to use numpy's own (less) random number generator instead of the secrets module (<https://docs.scipy.org/doc/numpy/reference/routines.random.html>).

Needless to say, Numpy is capable of storing much more than dice rolls and shuffling cards. If you want to store and manipulate other data types in an array, take some time to read through the Numpy manual (<https://docs.scipy.org/doc/numpy/>). This contains details on more advanced ways of managing data such as byte ordering and subclassing, which haven't been covered here. By default arrays are accessed as read only during runtime so changes you make are not saved from one session to the next. If you want to store an array in a file use `numpy.save` (there are more details at <https://docs.scipy.org/doc/numpy/reference/generated/numpy.save.html>). **LXF**



Quick tip

For more handy Numpy commands, use the DataCamp Cheatsheet available from <http://bit.ly/numpy-pdf>.



➤ You can use the Thonny IDE with Python 3.6 in Raspbian. Once you've installed the new version of Python, go to **Tools>Options>Interpreter** and select it from the drop-down menu.

On the disc

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

The best of the internet, crammed into a phantom-zone like 4GB DVD.

Distros

This month's lead feature is about

getting started with Linux. There are many differences between Linux and other operating systems that take some getting used to, but one of the major ones is the breadth of choice. From the hundreds of different distros through the dozens of different desktops to the countless choices of web browsers, mailers, editors and so on.

So which one of each to choose? Distros tend to only include what they consider to be good software in their default installations, so whatever your distro provides is a decent choice, but there are always alternatives. Don't be tempted to ditch a distro simply because you don't like the email program they provide, or the colour of the desktop. The nature of open source software means that a distro installation can only ever be a starting point that others think make sense. It's up to you where you go from there.

If you're new to Linux I suggest you pick a distro and stick to it, along with its default software, to start with. Once you have a good idea of what you do and don't like, start exploring the distro's software repositories. You never know what goodies you may find in there!



Neil

Perfect KDE starting point

64-bit

ROSA Desktop R10

We have two lightweight distros on the opposite page, so we should balance things up with something a little heavier on this side of the magazine. When it comes to full-featured, integrated desktops, the choice generally comes down to either Gnome or KDE. We probably have more Gnome-based distros (especially now that Ubuntu uses Gnome) on our DVDs these days, so let's even the score with a KDE desktop (*this really is a well-balanced DVD, try it! – Ed*).

ROSA Fresh is a Russian distro with a customised KDE desktop. As we're going to be a bit critical of KDE here, we should state that we are happy KDE users, so KDE diehards don't need to flame us. We feel that KDE is a good desktop in search of a decent configuration – the default is pretty awful.

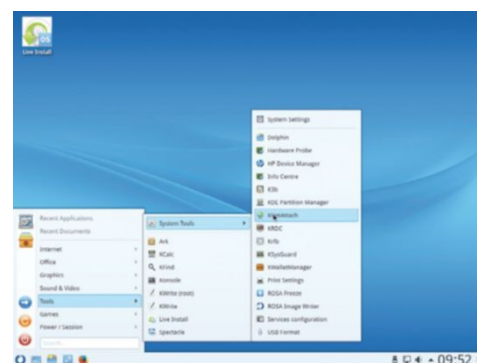
KDE is configurable to the nth degree and ROSA has taken full advantage of that to produce a slick and attractive KDE experience that gives a far better default starting point than the original. But it's still just a starting point. If you're going to stick with the default configuration of KDE, you really are missing out. It also produces a version with an LXQt desktop, but this is its flagship KDE release. If you want to see what LXQt is like, you can always install it from the ROSA repositories.

ROSA is intended for the more experienced Linux user. That's not to say you have to be a master of the command line or some sort of Linux

Jedi, and it's certainly not as entrenched in the deep end as Gentoo, but don't expect excessive hand holding. This is a distro for Linux users who want to try something new, something with very recent versions of the latest software.

ROSA has a good pedigree, being a fork of Mandriva. Mandriva and Mandrake were the most popular home distros for enthusiasts of all levels before Ubuntu took over top slot. So you have a solid base, up-to-date software and well-crafted configurations, so what else could you ask for? Well, there's some useful information on the wiki to help you get more out of it...

Login details: username **live**; no password.

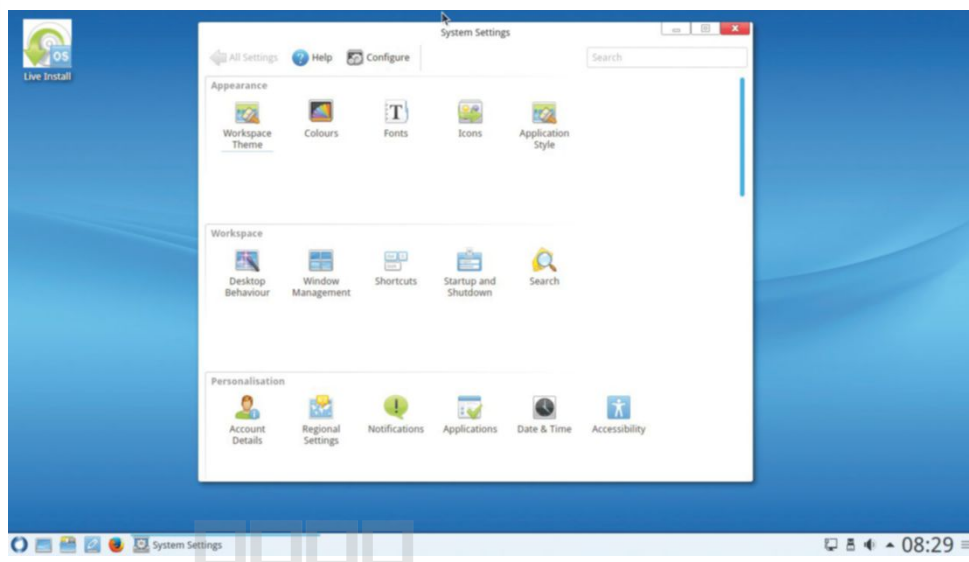


» A customised KDE desktop that has a more traditional look – something that many users want if the success of Linux Mint is any guide.

» Important NOTICE!

Defective discs

For basic help on running the disc or in the unlikely event of your *Linux Format* coverdisc being in any way defective, please visit our support site at www.linuxformat.com/dvdsupport. Unfortunately, we're unable to offer advice on using the applications, your hardware or the operating system itself.





New to Linux? Start here

- » What is Linux? How do I install it?
- » Is there an equivalent of *MS Office*?
- » What's this command line all about?
- » How do I install software?

Open [Index.html](#) on the disc to find out



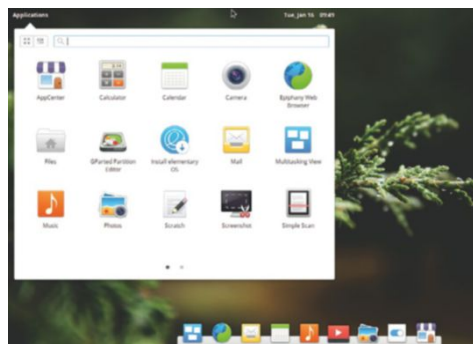
Distro for those new to Linux

64-bit

Elementary OS

There's not much to say about elementary OS that isn't covered in our main feature. It's meant to be a simple introduction to Linux for those familiar with other platforms. One of the things that confuses new Linux users is the amount of choice. If your OS has three text editors, two browsers and four media players, which one are you supposed to use? Elementary OS 0.4.1 makes it simple by including one tool for each job, yet still making alternatives available for when you want to go exploring.

Login details: username **elementary**; no password required.



Distro for old hardware

32-bit

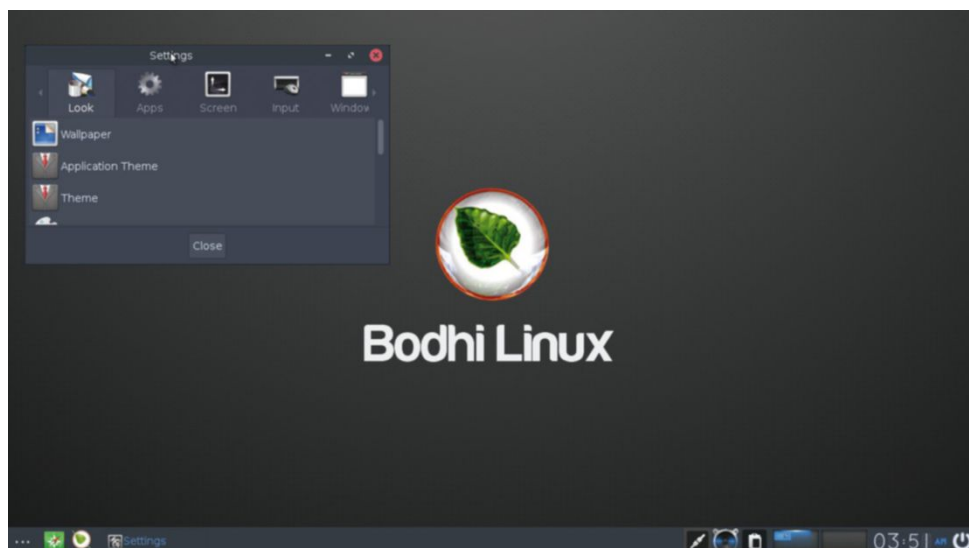
Bodhi Linux Legacy

We've included Bodhi Linux on the *Linux Format* DVDs before – it's a popular choice. Bodhi is one of a decreasing number of distros that still have a 32-bit version (Bodhi refers to it as "Legacy"). This makes it sound like a backdated version of the distro – not so!

The 32-bit version of Bodhi 4.4.0 contains the same software choices and versions as its 64-bit sibling, but built for 32-bit hardware. This means that owners of old hardware can still enjoy

a modern distro and software. There's clearly a performance penalty with less-powerful hardware, but Bodhi uses the fast and lightweight Moksha desktop, a fork of the popular Enlightenment, to give an excellent level of performance. That's not to say that Bodhi is only to be used on old machines, though. Install it on more modern hardware and it will really fly in a way that Gnome and KDE users can only dream of!

Login details: username **bodhi**; no password.



And more!

System tools

Essentials

Checkinstall Install tarballs with your package manager.

Coreutils The basic utilities that should exist on every operating system.

HardInfo A system benchmarking tool.

Kernel Source code for the latest stable kernel release, should you need it.

Memtest86+ Check for faulty memory.

Plopp A simple manager for booting OSes, from CD, DVD and USB.

RawWrite Create boot floppy disks under MS-DOS in Windows.

Smart Boot Manager An OS-agnostic manager with an easy-to-use interface.

WvDial Connect with a dial-up modem.

Reading matter

Bookshelf

Advanced Bash-Scripting Guide Go further with shell scripting.

Bash Guide for Beginners Get to grips with *Bash* scripting.

Bourne Shell Scripting Guide Get started with shell scripting.

The Cathedral and the Bazaar Eric S Raymond's classic text explaining the advantages of open development.

The Debian Administrator's Handbook An essential guide for sysadmins.

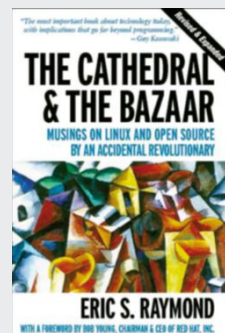
Introduction to Linux A handy guide full of pointers for new Linux users.

Linux Dictionary The A-Z of everything to do with Linux.

Linux Kernel in a Nutshell An introduction to the kernel written by master hacker Greg Kroah-Hartman.

The Linux System Administrator's Guide Take control of your system.

Tools Summary A complete overview of GNU tools.



Future Publishing Limited,
Quay House, The Ambury, Bath, BA1 1UA
Email linuxformat@futurenet.com

EDITORIAL

Editor Neil Mohr
neil.mohr@futurenet.com
Technical editor Jonni Bidwell
Art editor Elfrain Hernandez-Mendoza
Operations editor Cliff 'ImagineFX' Hope
Group editor in chief Graham Barlow
Senior art editor Jo Gulliver
Editorial contributors
Mats Tage Axelsson, Neil Bothwick, Nate Drake,
Kent Elchuk, Matthew Hanson, John Knight,
John Lance, Nick Peers, Les Pounder,
Mayank Sharma, Shashank Sharma,
Valentin Sinitsyn, Alexander Tolstoy, Mihalios Tsoukalos
Cartoons Shane Collinge

ADVERTISING

Media packs are available on request
Commercial director Clare Dove
clare.dove@futurenet.com
Senior advertising manager Lara Jaggon
lara.jaggon@futurenet.com
Advertising manager Michael Pyatt
michael.pyatt@futurenet.com
Director of agency sales Matt Downs
matt.downs@futurenet.com
Ad director - Technology John Burke
john.burke@futurenet.com
Head of strategic partnerships Clare Jonik
clare.jonik@futurenet.com

INTERNATIONAL LICENSING

Linux Format is available for licensing. Contact the
International department for partnership opportunities:
International licensing director Matt Ellis
matt.ellis@futurenet.com Tel: +44 (0)1225 442244

SUBSCRIPTIONS & BACK ISSUES

Web www.myfavouritemagazines.co.uk
Email linuxformat@myfavouritemagazines.co.uk
UK 0344 848 2852
International +44 (0) 344 848 2852

CIRCULATION

Head of newstrade Tim Mathers

PRODUCTION AND DISTRIBUTION

Head of production UK & US Mark Constance
Production project manager Clare Scott
Advertising production manager Joanne Crosby
Digital editions controller Jason Hudson
Production controller Nola Cokely

MANAGEMENT

Managing director Aaron Asadi
Editorial director Paul Newman
Art & design director Ross Andrews
Head of art & design Rodney Dive
Commercial finance director Dan Jotcham
Printed by Wyndeham Peterborough, Storey's Bar
Road, Peterborough, Cambridgeshire, PE1 5YS
Distributed by Marketforce, 5 Churchill Place, Canary
Wharf, London, E14 5HU www.marketforce.co.uk
Tel: 0203 787 9001

LINUX is a trademark of Linus Torvalds. GNU/Linux is abbreviated to Linux throughout for brevity. All copyrights and trademarks are recognised and respected. Where applicable code printed in this magazine is licensed under the GNU GPL v2 or later. See www.gnu.org/copyleft/gpl.html.

We are committed to only using magazine paper which is derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards. The manufacturing paper mill holds full FSC (Forest Stewardship Council) certification and accreditation.

Disclaimer All contents © 2018 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher, Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/ services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensee a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend, adapt all submissions.

All contents in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your systems, peripherals or software through the use of any guide. Many brain cells died to bring us this magazine.

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.



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

Chief executive Zillah Byng-Thorne
Non-executive chairman Peter Allen
Chief financial officer Penny Ladkin-Brand

Tel: +44 (0)1225 442244

Protect your privacy... Beat GCHQ!

As the UK plunges towards Chinese-levels of state sponsored surveillance, protect yourself online.

Handling exploits

On the back of Meltdown and Spectre we explore how Linux Kernel devs handle exploits and patch the holes.

Streamline your photos

Explore tools and *Darktable* to help you process RAW photos and optimise an ever-growing collection.

Ardunio explored

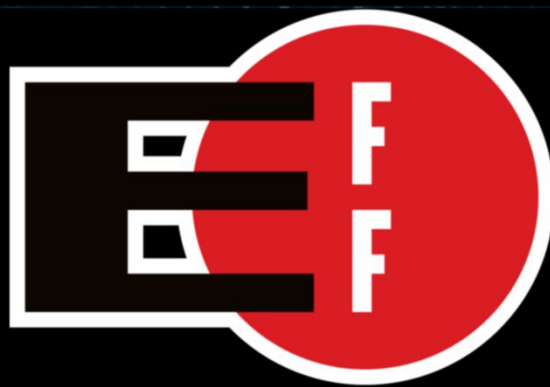
We tackle a new physical project so we can discover how to develop and build Arduino devices.

Get conferencing!

Bring people together and stay in touch with loved ones – we test the best open source conference tools.

Contents of future issues subject to change – we might have been asked to have a quiet “chat” with The Man.

LXF 235
will be on sale
Tuesday
13 March '18



The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier

LOSE YOURSELF IN A WORLD OF

Vinyl

FIND YOURSELF IN
OXFAM'S ONLINE SHOP

oxfam.org.uk/shop



OXFAM