

Crazyflie: Open source quadcopter

LINUX FORMAT

Get into Linux today!

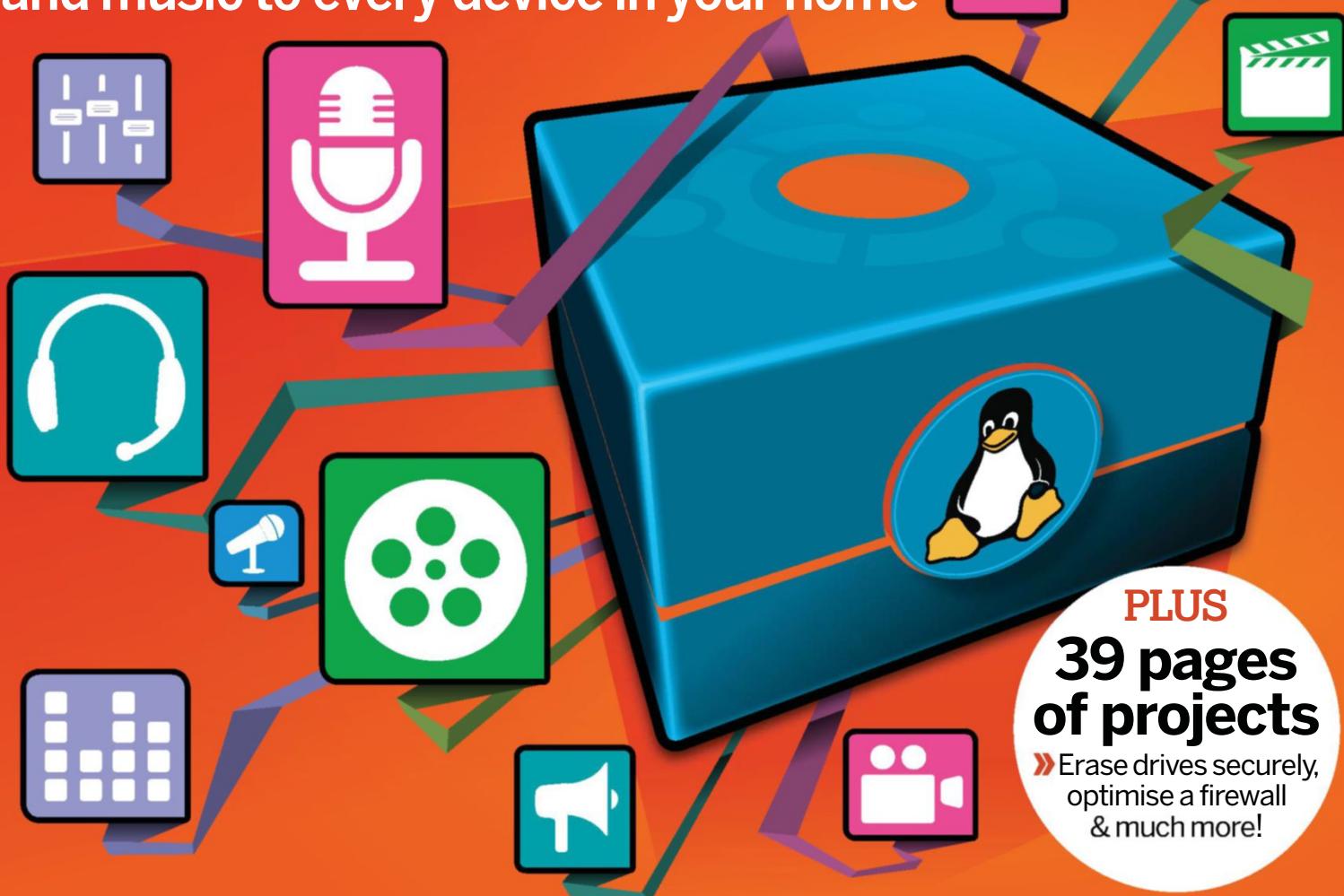
Speed up Python

» Use Cython to optimise your code



BUILD A LINUX MEDIA SERVER

Turn your PC into a hub – stream films and music to every device in your home



PLUS

**39 pages
of projects**

» Erase drives securely, optimise a firewall & much more!

Kernels

Explore Minix 3

» Try out the kernel that inspired Linux

Filesystems

Master ZFS & btrfs

» Set up a cutting-edge file system in your RAID array

Also inside

Mod Minetest

» Customise the Minecraft killer with your own content

Future

100% informed 70% recycled



No wonder
you paper

Did you know that almost 70% of paper across Europe is collected and recycled? In fact it's one of the most recycled materials of all.[†]

Magazines are printed on paper from natural and renewable wood which is all good to know if you love reading your favourite magazine.

† Monitoring Report of the European Recovered Paper Council, (ERPC), 2010

To discover some surprising environmental facts about print and paper, visit www.youlovepaper.info



Two Sides is an industry initiative to promote the responsible use of print and paper as a uniquely powerful and natural communications medium.

Print and Paper.
The environmental facts may surprise you



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 GPLv3.
- » We give you the most accurate, unbiased and up-to-date information on all things Linux.



Who we are

If Steve Ballmer got his wish and open source was wiped from the face of the planet, what would be the hideous consequences?



Jonni Bidwell

Well, we'd all be out of a job. But I don't know if open source would really stay dead for very long. Sooner or later someone would feel like sharing – whether or not there would be an internet to share on is another question. Like shipping containers, network protocols work best when there are open standards.



Neil Bothwick

It wouldn't be a problem. Just pull the previous version of Planet Earth from GitHub and fork it.

That's it.
Nothing to see here. Move along now.



Sean Conway

If open source software was wiped from the face of the planet, I would have to find something else to play with sitting in my basement. Like a well in the desert to a thirsty wanderer, open source is a selection of cool drinks for the pilgrim. Without open source, your beverage of choice is lost and you have to settle for what is on tap.



Andrew Mallett

Quite simply, the world as we know it would end without open source. The City of London, the financial centre of the world, runs on it. And even taking the world's finances out of the equation, how do we learn? How do we evolve? OSS is about sharing and evolution rather than only ever doing what you have always done.



Mayank Sharma

I can think of many implications (no Raspberry Pi, no Linux, no Apache, no world wide web) but the worst consequence for me personally (yes, I'm being selfish) would surely be that this fine publication wouldn't exist and I'd probably end up as visiting faculty at the Silsby Institute of Typewriter Maintenance.



A world without Linux

» Let's play a game: imagine a world with no open source software. Imagine Steve Ballmer invented a time machine, went back to 1953 and prevented the birth of Richard Stallman, the father of the Free Software Foundation. Overnight, everything open source vanishes from the face of the planet – but what changes?

Of course, GNU/Linux disappears. Something like 75% of the world's web servers grind to a halt or have to switch to Windows. You can kiss goodbye to every Android phone and tablet too. You're OK as you're an Apple owner? Nope, the Darwin kernel is based in part on the open source BSD kernel, while *Safari* uses the open source WebKit, to take just two examples of the open source elements that power both its mobile iOS and desktop OS X operating systems.

We'd be left in a world of Windows, but without iOS and Android to compete against, Microsoft would have been happy to continue flogging us all its Windows Mobile OS and Windows XP on the desktop. And with so many web technologies based on open source and open platforms, the internet as we know it would cease to exist: we've already kissed goodbye to *Safari*, but bang goes *Chrome*, *Firefox*, *WordPress*, Docker, OpenStack and OpenSSL – the list goes on and on.

This was just a silly academic exercise, but the point is to show how widely open source is used. It increases choice: Android can be adopted and adapted by any company. It speeds adoption of technologies: source code has to be made publicly available, so everyone can use it and contribute to it. It reduces costs: there's no need to develop technologies from scratch or buy them in at great cost, and tried and tested code can be reused. It fuels standards: Docker is a cloud phenomena that even Microsoft has to embrace.

So as you read this issue, taking in open source media centres, streaming standards, open server systems, *Minecraft* alternatives, filesystems, drawing packages, OpenLDAP, alternative kernels, programming and so much more, just thank Stallman* open source exists at all.

Neil

Neil Mohr Editor

» neil.mohr@futurenet.com

*We realise someone else would have championed the philosophy, but as well as katana-wielding Stallman? Never! <http://bit.ly/StallmanKatana>.

Subscribe today
See p32 for awesome deals

Contents

"Let us prepare to grapple with the ineffable itself, and see if we may not eff it after all."

Reviews

Google Nexus 6 17
The first phone with Android 5.0 Lollipop is a thing of beauty and a joy forever.

Google Nexus 9 18
In contrast, Google's first tablet with Lollipop is a solid, if unspectacular, performer.



► The Nexus 6 and 9: the Heaven and Hell of Android 5.0 Lollipop devices?

Samsung Chromebook 2 19
A Chromebook that is thin and attractive as well as affordable? What magick is this?

Puppy Linux 6.0 20
A fast, lightweight distro that will delight newcomers and users of older machines.



► Paws for applause: Puppy Linux is a fast, cute, beginner-friendly distro.

Cinnamon 2.4 21
The new version of this desktop rewards existing users, but holds few surprises.

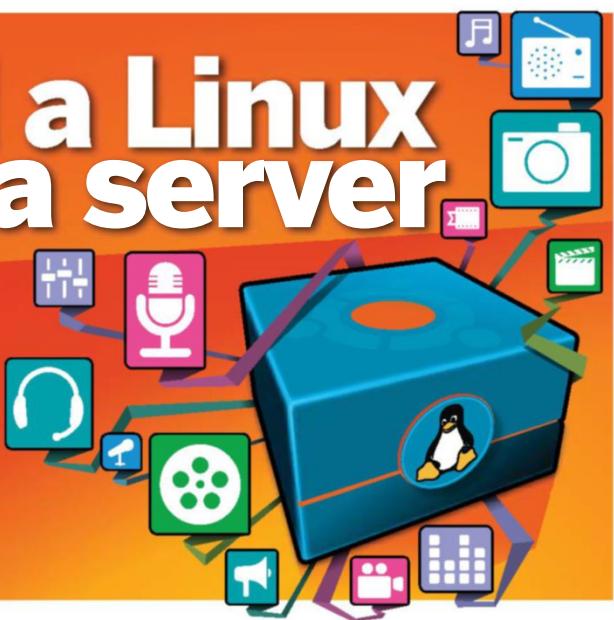
Vaultier 0.7 CE 22
Password managers are back in vogue, but is this a safe place to stuff all your secrets?

Hover 24
Add a touch of Minority Report to your PC with this inexpensive gesture-control board.

Games and books 25
We check out *Football Manager 2015* (meh) and *Adventures in Minecraft* (meh-ritorous).

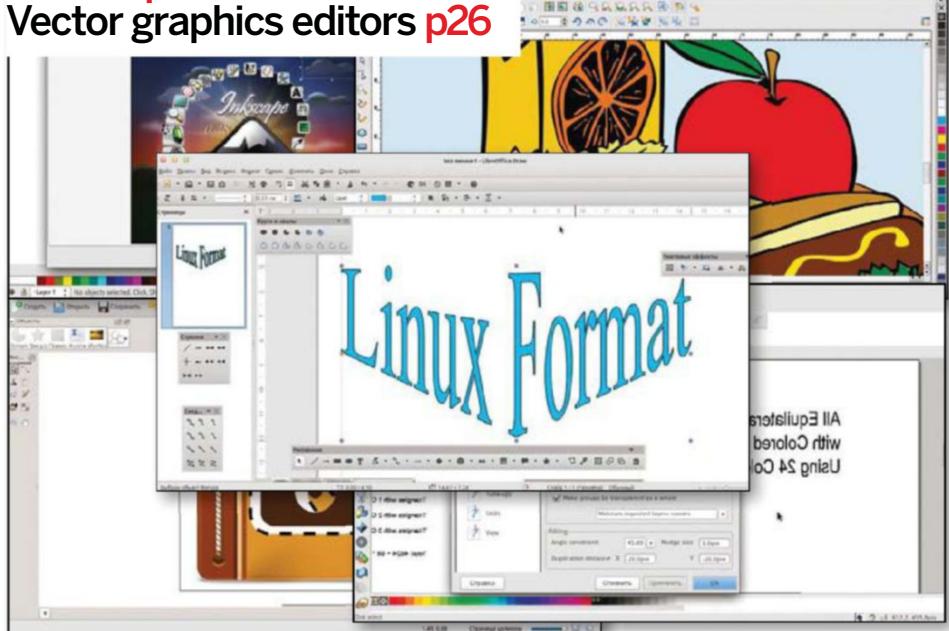
Build a Linux media server

Turn your PC into a hub to stream films and music to every device in your home p34



Roundup: Vector graphics editors

p26



Interview



“ The original version actually had the motors glued onto the circuit board itself! ”

The Crazyflie team discuss open source nanocopters p42



On your free DVD

Ubuntu 14.10

XBMCbuntu 13.2

OpenSUSE 13.2

» Killer distros for our cover feature.
PLUS: HotPicks and tutorial code p96



Treat yourself or a loved one to an LXF subscription! p32

Don't miss...

Master ZFS and btrfs 48

Pick from two of the most talked-up next-gen filesystems for your RAID array.



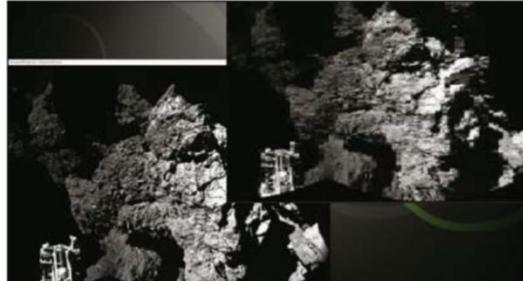
Mod Minetest 52

Fun with the open source *Minecraft*-inspired sandbox game using its API and many mods.

Coding Academy

Cython 84

Jonni Bidwell explains how to speed up your Python code, using an image-compression algorithm as a worked example (or an excuse to look at photos of the Rosetta mission).



R 88

Mihalis Tsoukalos provides an introduction to the statistical programming language in order to analyse system data. The things he can do with a box plot would make your eyes pop.

Regulars at a glance

News 6

Debian decides to keep *Systemd*, and the Raspberry Pi Model A+ ships. Plus: good news for Linux gamers.

User groups 10

Les Pounder samples the sweet treats of the Jam Packed roadshow.

Mailserver 12

Problems are solved, software praised and a few typos busted as the LXF team answers your emails.

Roundup 26

We challenge five top open source vector graphics packages to perform our devious series of tests without hesitation, repetition or pixelation.

Subscriptions 32

Our digital editions and print/digital deals are now available on Android!

Sysadmin 56

Dr Chris reveals the supercomputing secrets of the Met Office, and explains how to set up an OpenLDAP server to store user account details.

HotPicks 60

Alexander Tolstoy breaks out his asbestos gloves to bring you the hottest new open source software. Included this month: Lunar Linux, CherryTree, OpenSSH, Snapper, Remmina, Wal Commander GitHub Edition, rTorrent, Neverball, O.A.D., Konversation and Tomahawk.

Back issues 66

Buy now to assuage your guilt at missing them in stores. No collection would be complete without **LXF191**.

Next month 98

Explore bleeding-edge graphics and filesystems with our guide to setting up a next-generation Linux system.



» Our subscriptions team is waiting for your call.

Tutorials

HDDs

Erase drives securely 68

We test seven different methods for wiping confidential data from old drives: from **DBAN** and Secure Erase to a Lee-Enfield rifle!



» We don't know quite what made Darik so destructive but **DBAN** works.

ipset

Simplify your firewalls 71

Take the pain out of maintaining a firewall by using the ipset Netfilter extension to make configurations easier to read and modify.

Minix 3

Test a microkernel OS 74

Try out the kernel that inspired Linux as we explain how to get it running in *VirtualBox*.

Grub

Custom boot menus 78

Bend your boot menu to your will by customising its entries and appearance.

Awk, sed

Better text processing 80

Pull critical data from log files with our collection of power tips for text processing.



THIS ISSUE: Systemd debacle » New Pi board » OpenGL boost » Jolla tablet

INIT WARS

Debian keeps Systemd

The votes may have been cast but the debate about the init switch is likely to rage on as threats are thrown, resignations are tendered and a new distro is announced.

The argument over whether or not Debian should adopt *Systemd* as the default init system has been going on for some time now: we originally reported the disagreements back in [LXF183](#). Since then the controversy has escalated, with those on both sides of the argument getting increasingly irate.

Debian had been set on using *Systemd* as the default init daemon rather than an alternative such as *Upstart*, which had been gaining popularity among distros such as Fedora. Debian's governing technical committee decided to replace the *SysVinit* startup module with the more robust *Systemd* in the upcoming Jessie release of Debian.

SysVinit, critics claim that it's needlessly complex and suffers from feature creep.

The critics of *Systemd* had proposed a general resolution to the Debian community over the future of *Systemd*, but it failed. If the vote had succeeded, Debian software would have become 'init system agnostic', preventing the distro from being tied too closely to *Systemd*.

The debate has got so heated that the creator of *Systemd*, Lennart Poettering, has spoken about receiving death threats and other online abuse. Another Debian *Systemd* maintainer, Tollef Fog Heen, has stepped away from the project, commenting: "The load of the continued attacks is just becoming too much".

Colin Watson and Russ Allbery, key members of Debian's technical committee, have also resigned over the issue. In his

resignation announcement, Allbery claimed stress over community decision making caused him to step down: "Nearly every [technical committee] decision is now very fraught, and expressing those decisions, at least in the current framework, requires more skill, care, attention, and caution than I currently have mental or emotional resources to do".

Veteran developer Joey Hess also announced his departure: "We've made some good things, and I wish everyone well, but I'm out... If I have one regret from my 18 years in Debian, it's that when the Debian constitution was originally

"Devuan will be 'free from bloat as a minimalist base distro should be'."

This decision has split the Debian community, with a number of developers threatening to create a fork of the distro if *Systemd* was used.

On the campaign website <http://debianfork.org>, a group of self-titled veteran Unix admins commented: "We don't want to be forced to use *Systemd* in substitution to the traditional Unix *SysVinit* init, because *Systemd* betrays the Unix philosophy."

The philosophy in question states that "software should do one thing, and do it well" – a criterion they believe that *Systemd* doesn't satisfy.

While supporters of *Systemd* praise its versatility, and contend that it is a much-needed update to the archaic



Image credit: Harald Hoyer

Lennart Poettering and others working on the *Systemd* project have faced personal attacks by anti-*Systemd* campaigners.

proposed, despite seeing it as dubious, I neglected to speak out against it. It's clear to me now that it's a toxic document, that has slowly but surely led Debian in very unhealthy directions."

While Lennart Poettering and some of his co-workers on the *Systemd* project have decried personal attacks by anti-*Systemd* campaigners, they themselves have been accused of antagonising people who are against the change.

Their critics accuse them of acquiring a reputation for arrogance and inflexibility among open source developers; and they have also been accused of breaking compatibility among other software modules.

As this issue of *Linux Format* goes to press, the failure of the general resolution has lead the breakaway group of veteran Unix admins to announce Devuan (www.devuan.com), pronounced 'DevOne', which will "preserve init freedom" and will be "free from bloat as a minimalist base distro should be".

The team says it expects users to be able to switch from Debian 7 to Devuan 1 in "spring of 2015".



HARDWARE

New Raspberry Pi Model A+ released

The Raspberry Pi Foundation releases a more flexible Pi.

The Raspberry Pi Foundation has released a brand new model of its diminutive computer and – perhaps most crucially – kept the cost down to just \$20 (around £12.75). Dubbed the Model A+, the new Pi is the successor to the Model A, and in comparison to the Model A, increases the size of the GPIO header to 40 pins, adds a new micro SD slot, reduces power consumption and adds an improved audio circuit. This is all included in a unit 2cm shorter than the Model A.

Like the Model A, the A+ comes with only one USB port, 256MB RAM and no Ethernet connection, so if you're planning on creating more complex Pi projects, you'll probably want to stick with the Model B+, which comes with the same GPIO, but with four USB 2.0 ports and an Ethernet connector.



There's a new Raspberry Pi on the block. The Model A+, which replaces the Model A, costs \$20.

GAMING

Games graphics get speed boost on Linux

New driver boosts OpenGL performance on Haswell GPUs.

Do you play games on your Linux system using Intel's integrated Haswell GPU? If so, we've got some great news for you: LunarG (www.lunarg.com), a company dedicated to tuning 3D graphics drivers, has unlocked a significant graphics performance boost.

The company was contracted by Valve Software to find out why Intel's OpenGL driver for Windows was still generally faster than the open source Linux driver. LunarG's research discovered that there's a difference in how hardware samplers were working in each driver. LunarG passed these precious findings on to Intel who worked with hardware engineers to find the part of the Linux driver that needed to be changed to boost performance in games.

Once the driver was changed, to the delight of LunarG and Intel, performance improved drastically – and in some ways, the difference was better than either company had hoped. In the benchmarks and tests that followed, it was found

that *Left 4 Dead 2* ran around 17 to 25% faster and *Counter-Strike: Global Offensive* was boosted by between 16 and 20%. The next-generation lighting benchmark *Lightmark* also saw a huge 60% performance increase.

A new driver with these improvements will be made available soon, so Haswell owners (which at the moment is the only GPU that benefits from these performance increases) will soon have their games running better than ever before.



Games like *Left 4 Dead 2* will soon get a graphics performance boost from a better driver.

Newsbytes

» China is set to upgrade a significant number of PCs to Linux by 2020. We've previously reported that the Chinese government had banned the use of Windows 8 for government computers due to security concerns. Instead of the much-maligned Microsoft OS, Chinese governmental machines would run on a government-sanctioned version of Linux. It has now emerged that China could be seeking to move its computers to Linux by 2020. This date hasn't been confirmed; nor has there been an indication of what Linux distribution would be used, though the most likely candidates are Kylin, Ubuntu Kylin and Red Flag Linux.



Ubuntu Kylin is the most widely supported Chinese Linux.

» Jolla has started an Indiegogo campaign to create a crowdsourced tablet and in only a few hours has managed to smash its Indiegogo goal of raising \$380,000 (about £242,000 or AU\$437,000). At the time of writing, it had managed to raise over \$1.4 million. The tablet will run Jolla's own open source Sailfish OS 2.0 and sport a 64-bit 1.8GHz quad-core Intel processor, 2GB of RAM and come with 32GB of internal storage along with an SD card slot. The display is only slightly smaller than the iPad Mini 3 at 7.85 inches but matches the resolution at 2,048x1,536 for a pixel density of 330ppi.

» Scientists at the University of Utah (www.flux.utah.edu/project/a3) have created self-repairing software that can detect and destroy never-before-seen viruses and malware, as well as ensuring that the invader cannot infect the computer ever again. Known as A3 (which stands for Advanced Adaptive Applications), the software is designed to protect servers and enterprise-level computers running Linux. It works with a virtual machine and can be used to stop a virus, repair the damaged software code, and then learn to never let the bug infect the machine again. The software is open source and could be incorporated into commercial products.

Comment

Europe attacks

Michael Meeks

66

It was extraordinary this month to see the



European Parliament voting to try to encourage the splitting up of Google. In Europe, Google has a large market share – but then again, the service is excellent. At this stage, a quick disclaimer: Google supports The Document Foundation, has provided me with sample hardware for development, and gives huge support to open source projects via Google Summer of Code and elsewhere – but that doesn't mean it gets everything right. The community dynamics around Android, for instance, are highly dysfunctional.

Oh so fungible

Search, it seems to me, is the ultimate in easily substitutable products. I recall the overnight demise of my usage of Alta Vista – the 'search engine that helps you find exactly what you're looking for' just didn't seem to any more. More recently, given the wave of privacy concerns, I've tried DuckDuckGo, 'the search engine that doesn't track you', and it worked rather well, too.

So what barriers are there to a Google-beating search engine in Europe? The era of cloud computing means the need to first build a hyper-scale datacentre is no longer there; the necessary compute is a commodity available to any startup. In the USA, of course, there are questionable software patents, impeding innovation, but thanks to more enlightened European legislation, these are not a problem. Of course, scale provides economies and also attractiveness to advertisers – so significant capital would need to be ventured. Beyond that, are there truly network effects in search?

The sight of companies that have happily milked huge network effects themselves – not to mention having been caught abusing their resulting monopolies – attacking Google is rather amazing.

► Michael is a pseudo-engineer, semicolon lover, SUSE LibreOffice hacker and amateur pundit.

99



Hitting the mirrors

What's behind the free software sofa?

GUADALINEX 9

The latest in a rapidly growing line of Linux distros that have been developed by local governments. Guadalinex has been created by the Council of Economy, Innovation and Science of the Government of Andalucia. Designed to give all citizens of the Spanish province access to information technology, it's based on Ubuntu 14.04 with elements of Linux Mint and Debian.

It comes with the Linux kernel 3.13, Cinnamon 2.2.16, Firefox 24,

LibreOffice 4.2.6 and is available in two flavours: a man edition and a lite variant that has been built for older hardware and includes the LXDE desktop environment. It can be downloaded from www.guadalinex.org, though you might want to get out your Spanish phrasebook.



FEDORA 21

If all has gone according to plan, Fedora 21, the latest distro from the Red Hat-sponsored project, should be available to download as you read this issue. This update will feature a number of new and improved features, including a fully integrated application installer, the Cockpit Management Console, unified crypto policies, Make 4.0, Gnome 3.14, Headless Java, Java 8 and much more.

Find out more at <http://bit.ly/Fedora21Changes>.



► Fedora 21 is a big update that includes loads of new features.

MAGEIA 5

The latest major release for the popular Mandriva Linux fork should be available to download from

www.mageia.org. It will include the Linux 3.15 kernel, KDE 4.13.2, Gnome 3.13.3, Systemd 208 and other updates to the latest versions of open source packages. The btrfs



file system will be used by default, and an upgrade to Samba 4 and the network tool Drak will be improved. Phonon VLC will also be the default media player in this release.

ABIWORD 3.0

Popular word processing program *AbiWord* has been updated to version 3.0. The latest version benefits from a number of bug fixes, along with improved localisations for a number of languages. Paragraph borders and shading support have also been added to make documents look more impressive. RDF (Resource Description Framework) support

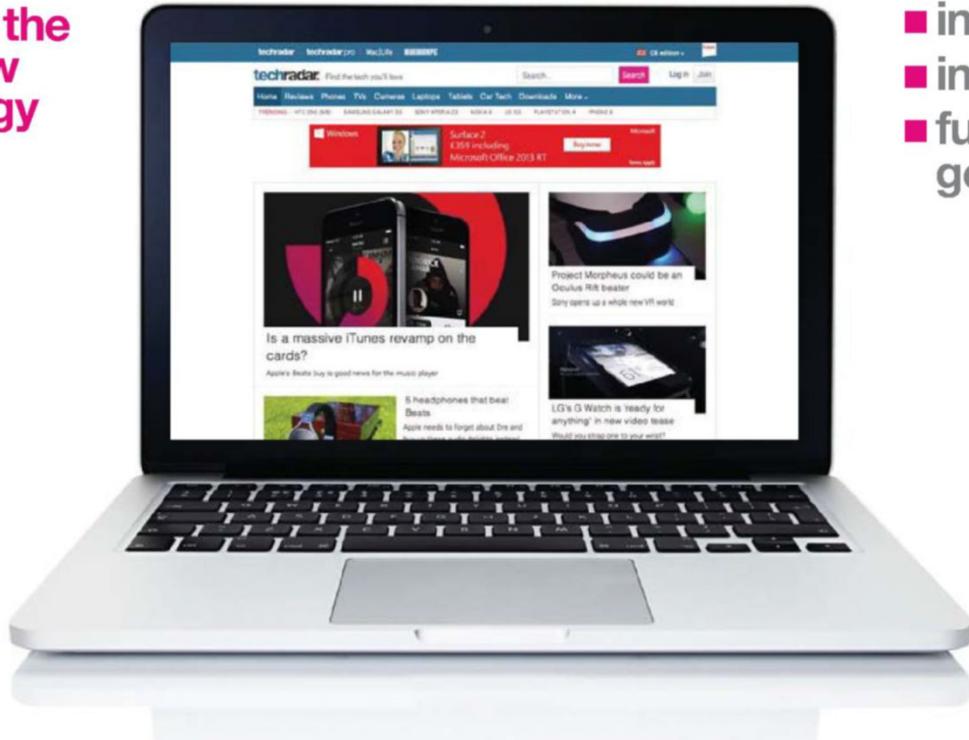
for ODT and ABW files has been added, and there is a new plugin for collaboration called Telepathy that enables you to share documents with your contacts. More information about the latest version of the program can be found at <http://bit.ly/AbiWordv3>.



Introducing a global tech brand
that promises to change the way
you consume technology

techradar.

Covering the
latest new
technology



- in depth
- in detail
- fully tested gear

techradar.
www.techradar.com



United Linux!

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

Find and join a LUG

» Bristol and Bath LUG

Meet on the fourth Saturday of each month at the Knights Templar (near Temple Meads Station) at 12:30pm until 4pm.
www.bristol.lug.org.uk

» Blackpool Makerspace

Meet every Saturday, 10am to 2pm, at PC Recycler, 29 Ripon Road, FY1 4DY.
<http://blackpool.lug.org.uk>

» Edinburgh LUG

Meet on the first Thursday of the month at the Southsider pub, West Richmond St.
<http://edlug.org.uk>

» Hull LUG

Meet at 8pm in Hartleys Bar, Newland Ave, on the first Tuesday of every month.
<http://hulllug.org>

» Lincoln LUG

Meet on the third Wednesday of the month at 7pm, Lincoln Bowl, Washingborough Rd.
www.lincoln.lug.org.uk

» Liverpool LUG

Meet on the first Wednesday of the month from 7pm onwards at the Liverpool Social Centre, Bold Street; and one Saturday mid-month.
<http://liv.lug.org.uk>

» Manchester Hackspace

Open night every Wednesday at their space at 42 Edge Street.
<http://hacman.org.uk>

» Surrey & Hampshire Hackspace

Meet weekly each Thursday from 6:30pm at Games Galaxy in Farnborough.
www.sh-hackspace.org.uk

» Tyneside LUG

Meet from 12pm, first Saturday of the month, at the Discovery Museum, Newcastle.
www.tyneside.lug.org.uk

Jam Packed

A new Raspberry Pi event that's travelling the UK.

Raspberry Jams have introduced lots of people to the Raspberry Pi and its fantastic community of makers and hackers. Started by ICT teacher and keen supporter of the new computing curriculum Alan O'Donohoe in 2012, the first Raspberry Jam was a means to an end: namely, obtaining access to the then rare prototyping boards. While Raspberry Jams are spreading across the world, there are some areas that have yet to sample the Pi, so Alan has come up with a new idea: one which will bring the experience to your doorstep and cater for many different groups of users.

His new project, Jam Packed, is a two-day event centring on the Pi and digital skills. Funded by the Department for Education and the Raspberry Pi Foundation, Jam Packed brings the Pi into a school or community space enabling children, teachers and parents to learn more about it.

Day one is called 'Hack To The Future' and is a mixture of teacher training and experiential learning for

pupils. Learning focuses on the many different languages and technologies available from the open source community, such as Python and Mozilla's suite of Teach the Web tools. In the evening of day one, there's a family hack jam, which aims to show families what fun computing can be.

Day two follows the tried and true recipe for a Raspberry Jam and involves everyone showing off their projects and helping others to be inspired to pick up the Raspberry Pi and make computer science more fun.

To learn more, visit Alan's blog (<http://teachcomputing.wordpress.com>) to find out when the Jam Packed circus will be in your town. **LXF**



The Raspberry Pi community includes makers and hackers who love to share their knowledge.

Community events news



OpenTech

June 2015 is the tenth birthday of this fantastic celebration of open source. OpenTech is a one-day unconference in London that showcases the best of open source. At the last event in 2013,

delegates learnt about the UK government's digital strategies, how we can use open data to help NHS trusts to manage cheaper alternatives to branded medicines, and more. A ticket is £5 on the door.
www.opentech.org.uk/2015

jQuery UK 2015

When looking for a JavaScript library, most people – including Google, GitHub and the UK's Government Digital Service – turn to jQuery. jQuery UK, now in

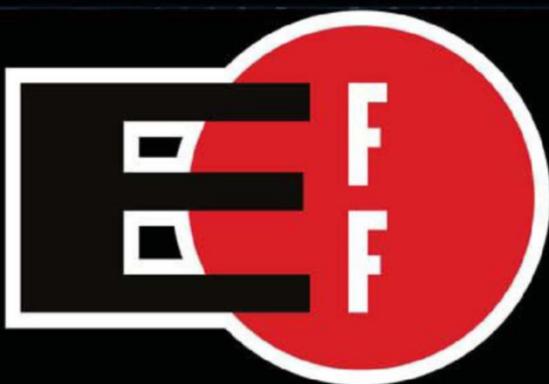
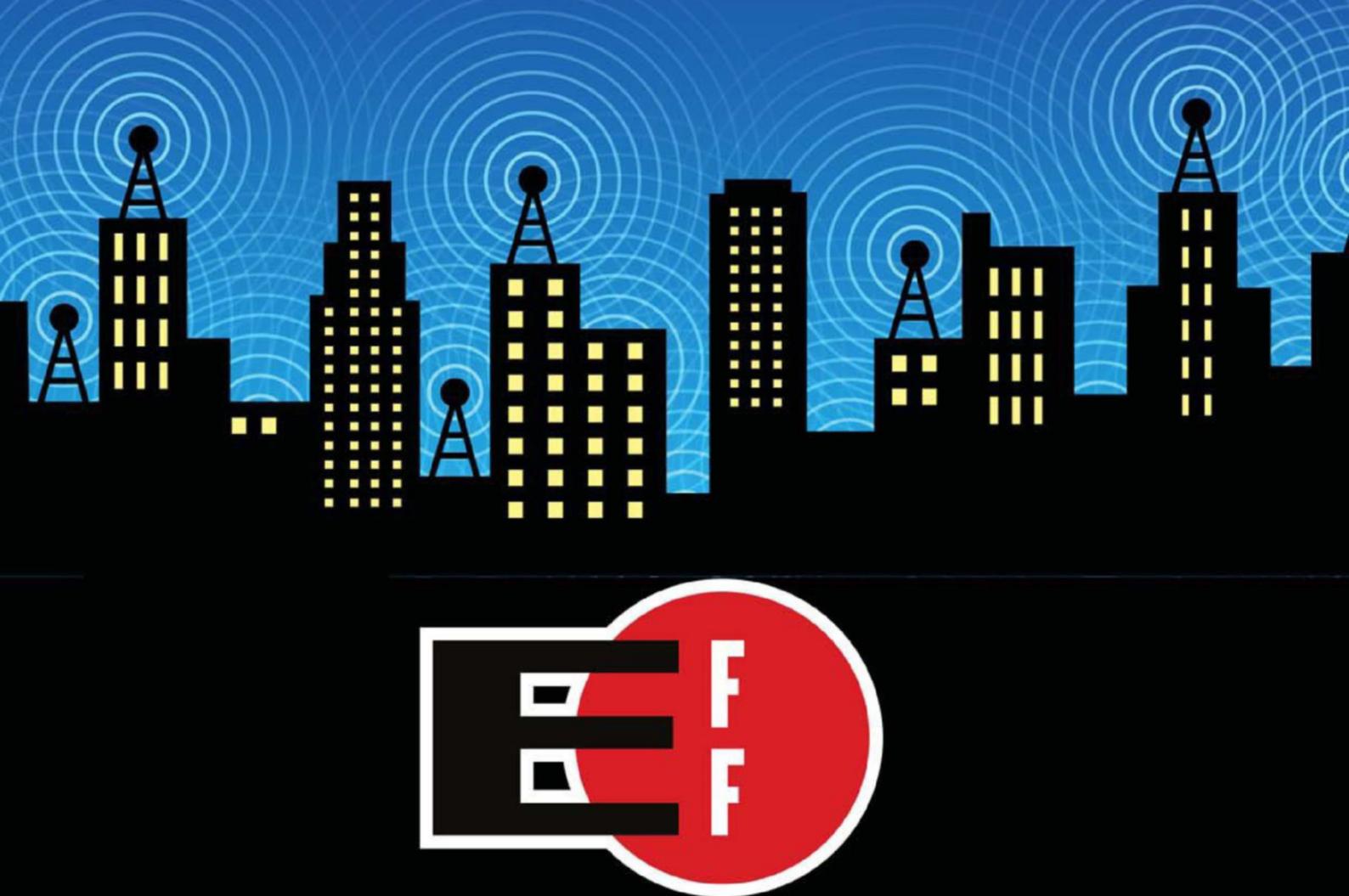
its fourth year, returns to Oxford for two days of front-end web development talks and workshops starting on 6 March. This popular event is supported by Mozilla and draws over 300 developers year on year. Tickets are £190 and more information can be found via the website.
<http://jqueryuk.com/2015>

Raspberry Jamboree 2015

The world's largest Raspberry Jam is back for 2015! Taking place in Manchester in March,

the Jamboree is your chance to learn more about the Pi via a series of talks and workshops led by the top people in the Pi community. In 2014, the Jamboree had talks on robotics, the unveiling of the Raspberry Pi display and hands-on sessions on learning Minecraft with the Pi. Details are still being worked on for this great event, and the best place to find out more is via Alan O'Donohoe's Twitter account.

<https://twitter.com/teknoteacher>



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

Mailservr



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

» Hacker theory

I recently picked up a copy of LXF188 and find it a treasure of useful information. I'm a newbie to Linux under the hood, though I have been using Ubuntu exclusively for home use since at least 2008. But while I love everything about Linux, I have had the misfortune to have been hacked, having surfed the net primarily over public unsecure Wi-Fi connections.

Using my laptop in the local public library I have been experiencing a few problems. My [connection] speed spontaneously changes from 56Mbps to 1Mbps. Headphones suddenly stop working. The browser screen greys out or closes after visiting certain news sites. Then there's the inability to run the `su` command. The inability to log into my laptop except at certain times. People and groups other than myself being given access to my laptop.

I would appreciate any information you can give me on securing my machine from intrusion. How can I find and erase a rootkit? What are the attack vectors for Ubuntu and Linux, and how can I protect against them? What services should be enabled/disabled?

I would also like to know where can I find the correct configuration files for my Linux distros. What should a 'normal'



Letter of the month

I Like Linux Lite, ya know

I've just got to congratulate the Linux Lite team on an excellent distro, for two reasons. First of all, Linux Lite is the first distro I've ever had that connected to my Wi-Fi router without any problems. I just had to enter the SSID of the router and the WPA2 key and got an instant connection. Second, Australia's biggest telco, Telstra, has just released a pre-paid dongle which it states is for Windows and Mac. I've been able to get one of these dongles working in the past, but only after a lot of help from the forums. This time I just plugged the dongle in to my Acer netbook with its Intel Atom processor and within one minute, the light on the dongle was flashing green, indicating that not only was it connected to the 4G network, but it was also downloading. I didn't have to do anything! I loaded up Firefox and went straight to the web browser's start page. I was a bit suspicious of this at first, thinking that maybe it was just a virtual page, but a quick session of



» Linux understands that not everyone wants a full-fat OS on their systems.

surfing proved everything was running OK. Well done Linux Lite!

Tony Moloney

Neil says: If you're wondering what the fuss is all about, head over to www.linuxliteos.com and see Linux Lite for yourself. We're glad you found the recommendation worthwhile, Tony. That's one of the joys of free and open source software: discovering amazingly useful projects.

config file look like, and what would distinguish it from a file that has been tampered with?

PW

Neil says: Wireless connection speeds can vary wildly on the fly, but I'm not entirely sure what the cause of some of your other issues is. However, regularly using

unsecured Wi-Fi connections is clearly a bad idea. A detailed guide to system security is beyond the scope of the Mailservr section, but we'd strongly suggest using a Live CD or USB-booting version of Tails (<https://tails.boum.org>), just to see if your many issues disappear. At least then you can

ascertain if your system is indeed being hacked or not.

» Speed daemon

The *Speed Up Linux* article [p32, LXF188] was very interesting, and I look forward to playing with some of the suggestions. It made me think

Helpdex

shane_collinge@yahoo.com





Both your virtual box and your real box are going to have to support VT-x for a fun virtual time.

of the fast-booting web-browsing OSes that laptop manufacturers started to come out with a few years ago.

Wouldn't it be great if there was a fast-booting basic Linux you could use like a kiosk browser? There are a few OSes that sound promising, like Kuki and Puppy, but when I tried a 64-bit version of Puppy (Fatdog) that supports UEFI, I was disappointed by the boot speed.

Vic5

Neil says: We aren't entirely sure what happened to those quick-boot OS options. Perhaps the added cost in the competitive motherboard market squeezed them out? Or perhaps the arrival of UEFI made them incompatible?

We aren't aware of any specific fast-boot Linux builds, but Jonni informs me a SSD install of Arch Linux can boot in as little as three

seconds. Alternatively, you could suspend to RAM. The system won't really be using much more power than when it's just powered off. TuxOnIce (<http://bit.ly/TuxOnIce>) implements just such a hibernate feature.

Hyper-V or VT-x?

In response to Rob Wolswijk's question about installing Linux in *VirtualBox* [Mailserver, p12, LXF190] you state that: "If you have trouble running the virtual machine, boot into the BIOS and make sure that the Hyper-V hypervisor setting is enabled."

I think what was meant is to make sure that the virtualisation extensions for the processor – VT-x for Intel processors and AMD-V for AMD – are enabled. Hyper-V is Microsoft's hypervisor product, and runs only on Windows. It's a minor typo, but it

could lead to some confusion. Also, I'm pretty sure the Execute Disable bit or NX bit usually needs to be enabled as well.

Mark Dean, via email

Neil says: Well spotted! Hyper-V is indeed Microsoft's hypervisor, and an entirely different kettle of fish to the VT-x and AMD-V BIOS settings.

Perhaps we had Microsoft on the brain thanks to all the love for Linux it's been showing recently: supporting it in Azure, cuddling up with Docker and taking .NET open source. It's a subject we'll return to later in Mailserver this issue, too.

Mythsterious

I run a Mythbuntu on my entertainment system, and I would like to see a tutorial on XBMC (or XMBCbuntu?).

I've tried to install and run it on my system to replace Mythbuntu, but the main issue that I find with it is getting the tuner to work. I have looked on various websites and also YouTube, but cannot get it to work. A tutorial on this would be appreciated.

On another note, for the last couple of issues I've received two issues. I'm hoping I'm only paying for one as that's all I subscribed to, but I've put the extra issues to use by putting them out at work for other people to discover.

Derek Martin

Neil says: There's no doubt that Mythbuntu is the king of TV tuner support. You might have noticed that we have a big feature on building a media centre around XBMC this issue (see p34), and while we don't cover tuner support in depth, the feature's author,

Nick Peers, does briefly discuss TV back-ends like MythTV. Perhaps it's one for a future issue?

As for the extra copies, our subscriptions department seems to have got into the Christmas spirit a little early this year. Thanks for putting them to good use!

Office alternatives

I write this email having trawled through your November 2014 issue (LXF190) and the DVD, but so far, I have still not found the relevant answers to the obvious difficulties that flow from changing operating system from Windows [to Linux].

As I have in the past tried to find these answers in the 'Linux press' with the same result, I suspect that the answers are either buried beneath the innumerable thickets of impenetrable jargon (isn't anything related to Linux written in plain English?) or that nobody has asked the questions before.

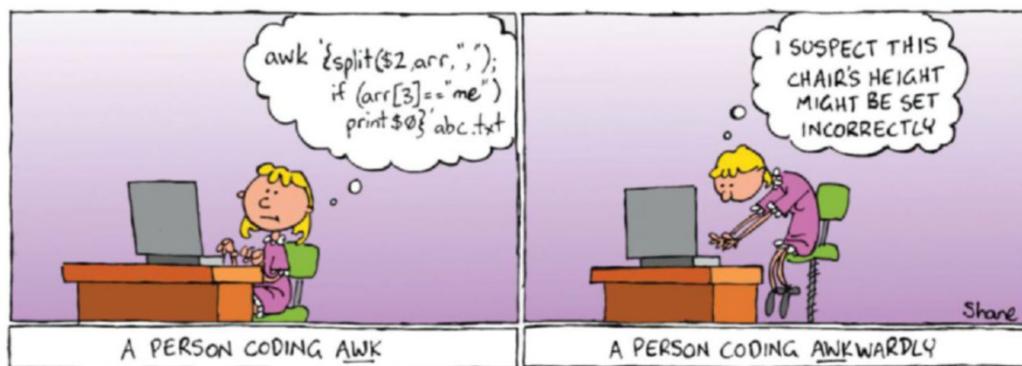
I'm looking for alternatives to the following: *Microsoft Office 2000*, *Microsoft AutoRoute*, *Microsoft Paint*, *Adobe Reader XI*, *BBC iPlayer*, Wi-Fi for the broadband hub and an image library. Is one able to reliably run these programs on Linux, or are there reliable Linux equivalents?

Yes, I have already tried the Windows version of *LibreOffice 4.2*, but that doesn't have a database that can output the selected data in a way that's readable in the Windows world. The prospect of having to re-key in over a dozen databases with over 20,000 records each doesn't exactly fill one with joy.

It may be that some of the programs are buried within the Linux operating system, but inserting your DVD results in a directory list including documents that are self-evidently not on the DVD, but elsewhere. This does not fill one with confidence to start any of the OSes from the DVD.

Andrew Procter

Neil says: It sounds like you haven't booted the **Linux Format** DVD, which is how it's designed to be used. This will usually take you into the Live CD version of the





» Microsoft might have fallen in love with Linux, but we know users that haven't.

» main distros on the disc, which vary from issue to issue.

For your database needs, *LibreOffice Base*, which is part of the *LibreOffice* suite, is a *Microsoft Access*-like standalone database. My understanding is this should connect to existing *Access* databases without issues, but the worst case would involve exporting a CSV file from *Access*.

There are also Linux equivalents for the other applications you mention. The latest build of Gnome includes the Maps map application; *Adobe Reader* 9 has Linux support, or you could try *MuPDF*; *Gimp* is a very capable image-editing package; and *Shotwell* is an excellent image library program. The BBC has switched to supporting HTML5 for iPlayer, and

has tested it in most common browsers on Ubuntu 11.x.

If you're looking for a plain-English guide for switching from Windows to Linux, you might like to check out the cover feature from LXF187, still available for purchase online via www.myfavouritemagazines.co.uk.

» PDFsam I am

I was just reading HotPicks in LXF191 and found *PDFsam*. This sounded a useful tool and I had an immediate use for it.

When I looked in the Debian repos, I found it there, so I installed it and it put an item in my Xfce menu. [It has a] basic interface and no help, but the article was very useful in explaining how to use it. I have a seven-page document which I

scanned in as seven PDF files. I loaded them in to merge and in about two minutes I had a single seven-page PDF. Brilliant. Incidentally, the PDF is almost exactly the same size as a ZIP file containing the seven one-page files – so much for compression! Give Alexander a thank you from me.

David Hignett, via email

Neil says: We're glad you're finding HotPicks to be of help! It's certainly the best place to find nuggets of open source gold.

» Cupertino love

First Microsoft promoted Linux by launching Windows 8 in all its shocking ugliness. Now Apple has decided to also promote the switch to Linux by releasing Yosemite, which is without doubt the ugliest interface ever. The internet backlash has begun, so I hope your subscription team is ready for the extra workload. I'd like to thank Microsoft and Apple for their efforts in encouraging people to adopt FOSS and Linux.

Trevor, via email

Neil says: Now, now. Microsoft simply decided to stumble into the user issues that the Gnome team discovered with Gnome 3. I'm not quite sure what Apple's excuse is, though: it seems to be taking OS X down a route of ever-increasing simplification, when the majority of people expressly seem to prefer the age-old desktop paradigm.

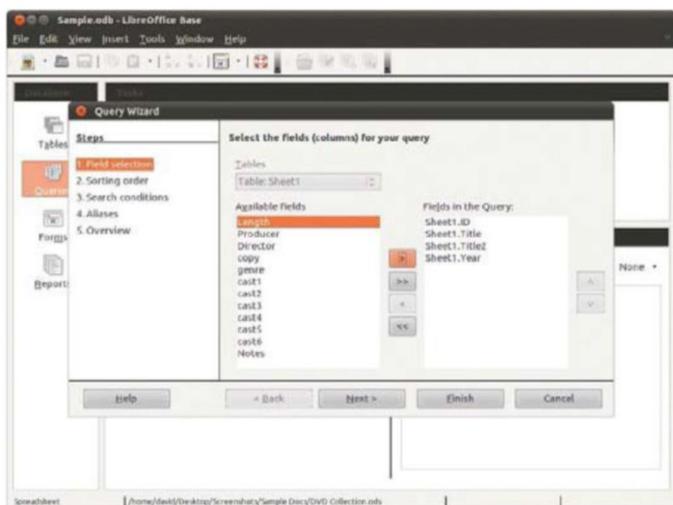
The beauty of Linux is that you can choose what you want: a simplified desktop like Gnome 3 or Unity, or a classic desktop like Mate or Cinnamon.

» More metadata!

Thanks for LXF192, which arrived through my letter box a few days ago. As usual an excellent magazine and DVD, but... erm, there's a problem on p69. It looks as if ExifTool (or was it MAT?) has thrown a wobbly and deleted some information at the end of the metadata article. Any chance you can tell me the ending?

Peter Lonsdale, via email

Neil says: Technical gremlins meant that the final two lines of Sean Conway's otherwise excellent article did not appear in the print edition of LXF192: digital subscribers should be unaffected. If you want to resolve our unintentional cliffhanger ending, you can find an updated PDF at <http://www.linuxformat.com/files/LXF192tutmeta.pdf>. LXF



» The *LibreOffice* community has created an amazing office suite that even the UK government has decided to adopt.

Write to us

Do you have a burning Linux-related issue that you want to discuss? Want to speculate about how we can read your minds? Then write to us at *Linux Format*, Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or alternatively send an email to: lxr.letters@futurenet.com.

T3



IN PRINT, ON TABLET AND SMARTPHONE

ON SALE NOW

Google Play

App Store

zinio

WWW.T3.COM

THE BEST

XMAS GADGETS

164 ESSENTIAL CHRISTMAS GIFTS FOR GADGET LOVERS

**TVs / Wearables / Audio Streamers / Headphones / Cycling Gear / Music / Boys' Toys
Tablets / Kitchen Gifts / Watches / Gaming / Fitness Tech / Outdoor Kit / AND MORE!**

WWW.MYFAVOURITEMAGAZINES.CO.UK/TECH-GADGETS/T3-MAGAZINE-SUBSCRIPTION/

THE ULTIMATE GIFT PACK

MINECRAFT

The ultimate gift for minecraft fans



This section contains three images related to the Minecraft gift pack. On the left is a copy of 'The Ultimate Guide To Minecraft' magazine, which is Volume 3 from Winter 2014. It features a Creeper on the cover and various articles about mods and maps. In the center is a smartphone displaying the cover of 'PC GAMER PRESENTS THE 50 GREATEST FREE GAMES OF ALL TIME', specifically the 'INDIE' section. On the right, a portion of the 'PC GAMER' magazine cover is visible, showing a character and the title.

ON SALE NOW

www.myfavouritemagazines.co.uk/GiftPacks

LINUX FORMAT

Reviews



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

Google Nexus 6

Matt Swider tries the Google phablet that's sweetened by Android Lollipop.

In brief...

» The first phone to gain Android 5.0 also happens to boast the best specification and biggest screen you've ever seen.

Specs

» CPU: Qualcomm Snapdragon 805
» Graphics: Adreno 420
» RAM: 3GB
» Screen: 5.96-inch 2,560x1,440 QHD AMOLED
» Storage: 32GB or 64GB
» Ports: Micro USB 2.0, Nano-SIM, 3.5mm audio
» Connectivity: 802.11ac 2x2 (MIMO), Bluetooth 4.1, NFC
» Cameras: Rear 13MP, Front 2MP
» Battery: 3,220mAh
» Weight: 184g
» Size: 82.98x159.26x10.06mm

» The Nexus 6 stacks up to be the best phablet we've seen.

The Nexus 6 was the first smartphone to run Android 5.0 Lollipop and there's no better way to experience all of the changes Google has made to its OS. It parades the new Material Design interface on a massive 6-inch display, moves seamlessly between apps thanks to the latest Snapdragon processor and 3GB of RAM, and lasts all day with a Qi-chargeable 3,220mAh battery.

The Nexus 6 has a head-turning 5.96-inch AMOLED display that's not only bigger and brighter than the Moto X, but edges out the 5.5-inch iPhone 6 Plus in every way. That's because it doubles down on Apple's full HD screen with a more stunning 2,560x1,440 resolution and 493 pixels per inch.

There are already members of the 'Quad HD' club, including Android members like the LG G3 and Samsung Galaxy Note 4, but Google Nexus 6 stands out from the pack with its exaggerated display.

There's no 6-inch juggernaut like it and its edge-to-edge display and on-screen soft buttons maximise the screen real estate while minimising the overall phone's dimensions.

The Google Nexus 6 also feels better in the hand for its size. Motorola's design, borrowed from the Moto X, gives it a sturdy aluminium metal frame and gently curved back. The curve feels natural against the hand – or, in most cases, hands – and the plastic back is at least smooth. The Nexus 6 colours

are limited to either Cloud White or Midnight Blue, but it has the advantage of being compatible with Qi wireless chargers: a feature tricked-out Moto X phones don't support.

Google's new phone also benefits from dual speakers at the top and bottom and comes with a Turbo Charger that juices up the phone with six additional hours of battery in just 15 minutes.

What's inside?

At the heart of the Nexus 6 is a 2.7Ghz Qualcomm Snapdragon 805 quad-core processor that has an Adreno 420 GPU. It's a top-of-the-line chip for Google's largest Android phone yet. Backing that up is 3GB of RAM and a reasonable 32GB or 64GB of internal storage: no silly 16GB model to cheapen the value.

All of these Nexus 6 specs aren't overkill: they're important for the lightweight, but feature-filled, Android Lollipop update that's pre-installed on the phone. Android does more things than ever in the background: running multiple apps, receiving notifications that hit the new lockscreen non-stop, making the instantly accessible Google Now available with one swipe to the left, and performing voice searches whenever you say the infamous words, "Okay Google". Around back, there's a 13-megapixel camera that seems to be on par with 13-megapixel Moto X snapper. It isn't. It has a better Sony IMX214 sensor with a wider f/2.0 aperture and optical image stabilisation.

The Nexus 6's US release date was 12 November and its pre-orders sold out immediately. The UK release date was very confusing and unclear until very recently. The Nexus 6 was meant to be available for preorder on Google



» The metal frame is a huge improvement on the Nexus 5.

Play during November, but that just hasn't happened. However, as we go to press, the Nexus 6 has popped up via O2 at £479 for the 32GB model: a fair price given the specs and internal storage, and that it's £499 on Google Play. This is also the same price as the 16GB model of the 4.7-inch iPhone 6, which is smaller in both physical design and storage size. **LXF**

LINUX FORMAT Verdict

Google Nexus 6

Developer: Motorola
Web: www.google.com/nexus/6
Price: £499

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

» Its tremendous display, premium specs and debut of Android 5.0 make the Nexus 6 the best phablet to date.

Rating **9/10**



Google Nexus 9

Google abandons its usual low pricing to enter the high-end tablet market. Matt Swider wonders if any Apple users will jump ship to the Android fleet.

Specs

- » CPU: 64-bit 2.3GHz Nvidia Tegra K1
- » Graphics: Kepler DX1
- » RAM: 2GB
- » Screen: 8.9-inch IPS LCD, 2,048x1,536 pixels, Corning Gorilla Glass 3
- » Storage: 16GB or 32GB
- » Ports: Micro USB 2.0, 3.5mm audio
- » Connectivity: 802.11ac, Bluetooth 4.1, NFC
- » Cameras: Rear 8MP, Front 1.6MP
- » Battery: 6,700mAh
- » Size: 154x228x8mm
- » Weight: 425g (Wi-Fi), 436g (LTE)

Google's Nexus 9 has been designed by HTC to be the Goldilocks of pure Android tablets and, for the most part, it succeeds at being 'just right' next to anything but an iPad. It's not as big as the seriously outdated Samsung-made Nexus 10 and not as small as the ASUS-crafted Nexus 7. It's the silver bullet or tablets – but it will cost more of your hard-earned gold.

Its Android competition includes the Samsung Galaxy Tab S, which flanks Google's 8.9-inch option with 10.5- and 8.4-inch sizes, and the Sony Z3 Tablet Compact. But more than hardware, what Nexus 9 has going for it is the fact that it's so far the only way to dive into Android 5.0 Lollipop on a tablet. That makes it a sweet enough product, in more ways than one. The new unified look, 'Material Design', is bright and colourful within the operating system as well as Google's own apps.

Considering that it comes from the company that brought us the polished-looking HTC One M8, it's no surprise that the Nexus 9 includes a brushed metal rim, but it sticks with a soft, rubberised back cover – the same as on the smaller Nexus 7.

The Nexus 9's 8.9-inch display is a little smaller than the 9.7-inch iPad Air 2, but offers the same maximum resolution on its IPS LCD screen.

The physical buttons on the rim are not always easy to find – if you've used the Nexus 6, you'd appreciate a power button accented with ridges. Thankfully,



» The first tablet with Android 5.0 Lollipop is a solid if not spectacular offering.

it's not always imperative to find that itty-bitty power button when the tablet is lying flat on a desk: a new 'double-tap to wake' feature in Android 5.0 Lollipop conveniently wakes the Nexus 9.

The lack of a micro SD card slot is the biggest design omission. There's no expandable storage whatsoever, meaning the 16GB model is going to be a tough sell if you use even a little bit of non-streaming multimedia.

Processing power

The Nvidia K1 Tegra processor is a switch from the typical Qualcomm Snapdragon chipset we're used to finding behind Android tablet displays, but the good news is that it's still a 64-bit system on a chip. The new Android Lollipop takes advantage of such 64-bit architecture, which makes the Nexus 9 a fair bet for the future. The processor and Lollipop together should result in more powerful and useful apps.

The Tegra K1 processor's single-core Geekbench 3 score actually surpassed that of the new iPad, averaging 1,939, while one core of the iPad's processor averaged a score of 1,815 in similar tests – although, of course, the Tegra K1 is a dual-core processor, as compared with the iPad's three-core.

At full brightness, our Nexus 9 battery tests found that a 90-minute

Full HD video took the battery life down to 82% – that's a drop-off that the iPad Air 2 just didn't match. In other real-world testing, the Nexus 9 lasted a day and a half before we needed to recharge it. Battery life is less of an issue on a tablet than a smartphone, but the Nexus 9 is no slouch.

If you're looking for a naked Android tablet, the Nexus 9 performs well and comes with some really premium touches to make it one of the best around. However, it's not the winner in any category at this point, so it will be interesting to see how Google uses this base model to improve the entire tablet ecosystem in time. **LXF**



Features at a glance



Key points

The keyboard case (sold separately) connects via Bluetooth and uses NFC to easily pair up.



QHD display

It's high res, but colour reproduction and contrast are distinctly average. HD movies aren't impressive.



Verdict

Google Nexus 9

Developer: HTC
Web: www.google.com/nexus/9
Price: £319

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

» An upgrade to last year's Nexus 7 and a great debut for Lollipop, but it doesn't measure up to the iPad Air 2.

Rating **8/10**

Samsung Chromebook 2

Samsung bucks the Chromebook trend, creating a uniquely attractive and thin laptop with a fanless design that's ahead of the curve, says **Kevin Lee**.

In brief...

» The Samsung Chromebook 2 is simply the first leap into what will likely be the future of all Chromebooks, and it's a bright first attempt.

Specs

- » CPU: 2.58GHz Intel Celeron N2840
- » Graphics: Intel HD Graphics
- » RAM: 2GB DDR3L
- » Screen: 11.6-inch, 1,366x768
- » Storage: 16GB solid state drive
- » Ports: 1xUSB 3.0, 1xUSB 2.0, HDMI, Micro SD slot, headphone/mic jack
- » Connectivity: 802.11ac Wi-Fi, Bluetooth 4.0
- » Camera: 720p HD webcam
- » Battery: 4,080mAh
- » Weight: 1.2kg
- » Size: 290x204x17 mm

When Samsung announced its second Chromebook, it made our heads turn. Rather than following the flock of Chromebooks with cheap plastic bodies, Samsung opted to give its new cloud-based laptop a leatherette finish on its lid, plus a 1080p display for the 13-inch model. Unfortunately, Samsung saddled the laptop with its Exynos ARM processor, designed for tablets, and it chugged when asked to multitask.

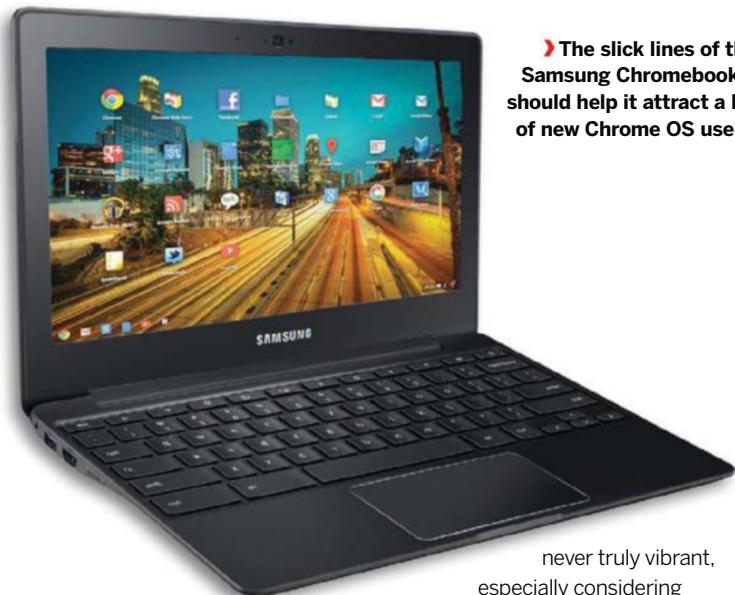
Five months later, Samsung has given the smaller 11.6-inch Chromebook 2 a refresh, swapping the mobile processor for an Intel Celeron CPU. Let's see if this quick fix has turned the Samsung Chromebook 2 into the winner it was meant to be.

The Chromebook 2 looks to be even better engineered than many mid-range laptops. The back of the laptop lid is wrapped with a leather-like finish that reminiscent of the Samsung Galaxy S5's design, down to the faux stitching.

The rest of the laptop's body is made of a light grey plastic, but does not flex even when you're mashing away at the keyboard. This is largely due to the laptop's metal internal frame, which extends across the entire body. Each of the corners is also reinforced with metal, which adds an extra bit of protection in case the laptop falls.

You're also looking at one of the thinnest Chromebooks yet, measuring in at 290x204x17mm. Most 12-inch Chromebooks hover around 1.3kg, but the Samsung Chromebook 2 wins over most, coming in at 1.2kg.

These specifications should look familiar if you're comparing it with other Chrome OS-based options, like the Dell Chromebook 11 and Acer C720. The Samsung comes with a very similar 11.6-inch and 1,366x768-pixel screen, 16GB SSD and 2GB of RAM. Despite these similarities, the new 2.58GHz Intel Celeron N2840 processor is the showstopper. Not only does it replace the Exynos chip, it's also Intel's fanless Bay Trail design.



» The slick lines of the Samsung Chromebook 2 should help it attract a lot of new Chrome OS users.

never truly vibrant, especially considering Samsung's penchant for oversaturated AMOLED panels, but they are accurate. What's more, you can actually see blacks as they were meant to be seen rather than gray blotches.

In our tests, the battery lasted for 7 hours and 32 minutes: very good, although still not quite enough to beat the Acer C720 with its absurd 8 hours and 30 minutes of run time.

Overall, the new Samsung Chromebook 2 is an improvement on the original. Though there are still performance niggles, it's definitely ahead of the curve. **LXF**

Still some lag

We also experienced some noticeable lag in Chrome. Oddly enough this stuttering occurred when trying to type a website or search query into the address bar. The Intel chip also sometimes chugged as it scrolled through a Google Drive document, especially with any documents containing tables or graphs. Otherwise, the Chromebook 2 had no trouble loading up 25 browser tabs while listening to tunes over Google Music. The Bay Trail chip also had no lag issues when playing 1080p video on YouTube.

Another fault of Chromebooks is that they all almost universally have terribly washed-out displays, save for a few outliers like the Chromebook Pixel. The Chromebook 2 manages to buck this trend as well – but not entirely – with a bright display that manages to be good but not great. Colours are

LINUX Verdict

Samsung Chromebook 2

Developer: Samsung
Web: www.samsung.com
Price: \$249.99 (UK £TBC)

Features	9/10
Performance	6/10
Ease of use	9/10
Value for money	10/10

» The Samsung Chromebook 2 is the first leap into what will likely be the bright future for all Chromebooks.

Rating **8/10**

Puppy Linux 6.0

Must all Puppy Linux reviews reference a dog? **Shashank Sharma** attempts to break the norm as he paws with the latest release of man's friendliest distro.

In brief...

» A lightweight distro that's aimed at older machines and systems with low resources. It features several custom graphical tools to help configure the system making it ideal for new Linux users. See also: Tiny Core Linux, Lubuntu.

Perhaps it's the name, or maybe it's the team of developers, but Puppy Linux seems to have mastered the art of being likeable. The lightweight distro has picked up several tricks since its initial release 11 years ago, but it's still fun, fast, friendly and reliably robust.

Codenamed Tahrpup, the latest Puppy Linux is shipped as a live, installable image. The project offers two 199MB images – PAE and non-PAE. The former is for machines which have more than 4GB RAM.

The distro uses *JWM* as its window manager and provides quick access to all the apps, which are all neatly filed away into categories, such as Personal, Utilities, Internet and Multimedia.

While most lightweight distros are usually also minimalist, Tahrpup sometimes features apps that provide extra functionality. This is most evident under the Internet and Multimedia menus, where you'll find several file sharing apps and music players.

The distro ships with the *Sylpheed* email client and the Firefox-based *Pale Moon* is the default browser. While Flash isn't installed by default, the browser is equipped to serve streaming videos. You get *AbiWord* and *Gnumeric*, by default, to work with office documents. The distro also features the *Pup Advert Blocker* utility for ad-free browsing.

The distro provides quick installation of popular apps. You can use the *GetFlash* utility to install the official *Adobe Flash* plugin. You can also

download and set up *LibreOffice* using the quick install utility from under the Document menu.

Despite its small size, the distro is full of applications, some of which are unconventional. It features *HomeBank*, to help you manage finances, *GWhere* to catalogue disks, *Figaro's Password Manager* and

graphical tools to manage *Samba* shares and setup firewalls.

If you find your favourite application missing from the impressive list of defaults, the binary compatibility with Ubuntu 14.04 means that you can use its package management tool to access Ubuntu's software repositories and install additional packages with ease. However, the *QuickPet* utility can be used to quickly install some of the most popular apps.

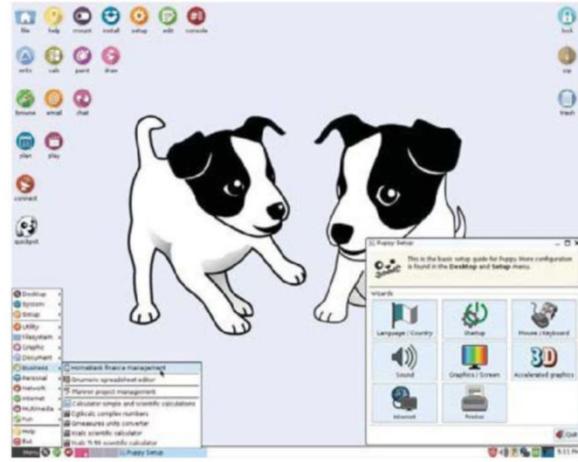
Configuring Tahrpup

The default Puppy desktop isn't internet ready. You're asked to configure the network when you first run Puppy and it provides several custom graphical tools to help you through the process.

Puppy features several custom tools with graphical wizards to help you configure the network, graphic display and printer and so on. The tools are newbie-friendly and the graphical wizards offer detailed instructions to help guide you along the way.

When you decide to end the live session, regardless of whether you run Puppy off USB or CD, it will prompt you to save the changes. If you decide to do so, Puppy will store all your settings, preferences and changes to the system to the specified path.

The distro ships with the *GParted* partition management tool, which comes into play if you decide to install Puppy to disk. The wizard-driven



» One of the fastest distros we've come across. It's also the neatest, cutest and... oh, stop wagging your tail!

installation is one of the easiest to follow and offers plenty of information at each step to help you decide how to proceed. Tahrpup uses *Grub4dos* as its bootloader and lets you make changes to the configuration files, if needed, during the installation.

The distro provides several icons on the desktop to some of the frequently used apps. Instead of the app name, however, the icons are named after the function required. For example, the write icon will open *AbiWord*, while play opens *VLC*.

Despite our determined efforts, we failed to find any fleas in this pup. The distro is one of the easiest to use and install regardless of your skill level. We'd definitely recommend Puppy Linux to new users looking for a fun distro. **LXF**

Features at a glance

Custom tools

You can easily configure the various aspects of the system from the Setup applications menu.

App collection

The distribution caters to a wide variety of users and features all the commonly used applications.

LINUX FORMAT Verdict

Puppy Linux 6.0

Developer: Puppy Linux team
Web: www.puppylinux.org
Licence: GPL/LGPL

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

» One rarely comes across such a perfect little distro. The custom tools simplify the configuration process.

Rating 10/10

Cinnamon 2.4

The fork of Gnome 3 Shell has matured into a standalone desktop environment of its own. **Mayank Sharma** checks out the latest iteration.

In brief...

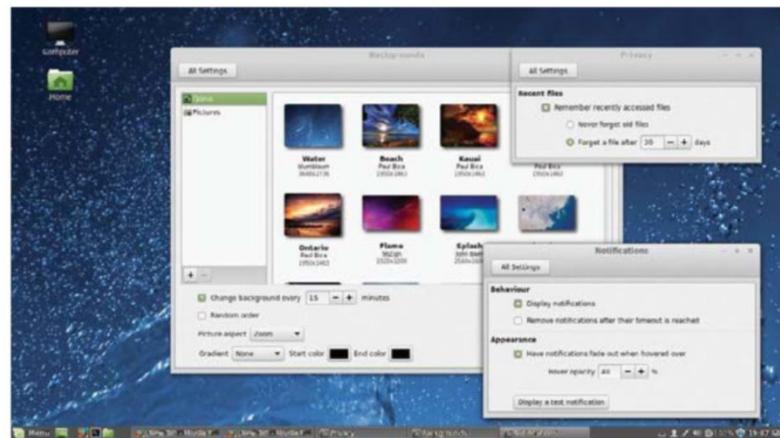
» A GTK3-based desktop forked from the Gnome Shell. See also: Gnome 3 and elementary OS's Pantheon.

The Cinnamon desktop has come of age. In addition to Linux Mint, which leads its development, the desktop environment is popular among users of other distros who want a desktop that's based on the latest libraries but sticks to the traditional desktop metaphor. In three years of existence, Cinnamon has grown from a fork into a standalone desktop of its own, which has been adding new features and functionality with every subsequent release.

With all the core components in place and its usability polished to a buff, the developers are now focusing on enriching the user experience. The latest release includes many features that the desktop has borrowed (with proper attribution) from other popular desktop environments and even from proprietary operating systems.

For example, the new release features a revamped Background Settings window that can now cycle through background images, as they do in Windows 8. This feature also brings along a new applet that you can add to your panel to control the background slideshow. There's also a new extension called nemo-emblems which enables you to add emblems to any content displayed by the Nemo file manager. Nemo itself features an improved sidebar, a redesigned toolbar and gets a new context-menu item to open a terminal in the current directory.

Also, as you'll find in Windows, the Super+E keyboard shortcut has been



» You can give Cinnamon 2.4 a go in Linux Mint's 17.1 Rebecca release.

added to launch the home directory. The desktop supports single-button touchpads like the ones on the MacBook and you can customise the actions for multi-finger taps. The desktop also borrows the zoom launch effect from the Gnome Shell and the Network Settings window is based on Gnome's as well.

Spring clean release

Cinnamon 2.4 has several UI improvements: a number of modules in the desktop's Control Panel have been redesigned for easier operation, such as Theme Settings which now has a new theme selection mechanism. The Panel Settings now let you place multiple panels on the desktop and you can control the size of their icons and text individually. There's also a new Privacy Settings window, where you can ask the desktop not to keep a record of recently accessed files or purge the list after a preset duration. You can also control the appearance and behaviour of notifications with a dedicated section.

The developers have also made some background changes, reviewing the code and its various components with static analysis tools, which has fixed several bugs as well as some 30 memory leaks. This has caused several minor refinements and changes, such as the removal of the timeout in the logout window. Furthermore, you can configure compositing in full-screen mode without restarting the desktop,

and the developers have tweaked the code of the composition manager to improve performance of games.

At first glance, Cinnamon 2.4 doesn't look all that different from the previous release, and if you haven't experienced any performance issues with previous releases, you won't notice any major performance gains with this one. But the developers have been toiling hard to iron out the unseen issues in the underlying code and introduce some minor but much requested features in the foreground. Cinnamon 2.4 isn't designed to attract new users but reward existing ones. In Linux Mint tradition, the desktop is released a month or so before it's included in the latest release, which for Cinnamon 2.4 was Linux Mint 17.1, so it's available to download now. **LXF**

Features at a glance

Revamped settings

A couple of modules in the System Settings Control Center were redesigned and new ones added.

Polished UX

Besides the usual bug fixing, some components were tweaked to create a smoother experience.

LINUX FORMAT Verdict

Cinnamon 2.4

Developer: Linux Mint team
Web: <http://cinnamon.linuxmint.com>
Licence: GPL v2

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

» A regular update to the popular desktop environment that further refines the existing experience.

Rating **8/10**

Vaultier 0.7 CE

Being the keeper of secrets can be challenging, **Shashank Sharma** discovers, as he tries out collaborative password manager *Vaultier Community Edition*.

In brief...

» A collaborative password manager and file storage tool that's designed for a very specific class of user and requires precise system set up.
See also:
Teampass,
MyPasswords.

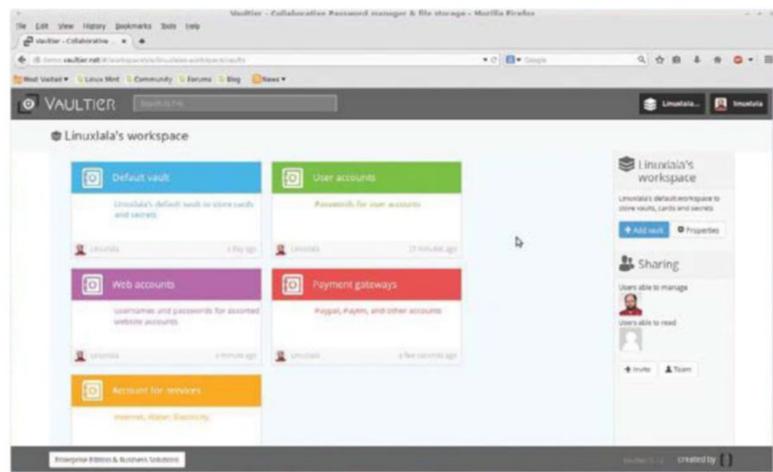
After a lull that lasted several years, and saw many existing tools retired from active development, password managers are back in vogue. The new batch, including *Vaultier*, now boast features such as file storage and collaboration.

The *Vaultier* project offers two variants, a BSD-licensed Community Edition and an Enterprise version, which comes with three different fee-based subscription plans.

For the Community Edition, the project provides three installation modes. You can install it as a Docker container or using *pip*, the Python installer, and Ubuntu users can use the provided shell script (the script is intended for Ubuntu 14.04 Server).

Regardless of the installation mode you choose, the project requires extensive configuration. *Vaultier* requires a PostgreSQL database, *Nginx* web server and uses uWSGI to run the app. The documentation walks you through the configuration of each of these but it's a very involved process and you're required to muck about with configuration files manually. The vanilla documentation provides only the minimum of instructions and you're on your own if you decide to configure *Vaultier* to work with *MySQL* or *Apache*. The project doesn't even have a mailing list, forum or a troubleshooting FAQ to help you resolve installation issues.

Vaultier bills itself as highly secure and provides key-based RSA encryption. When you first create an



» **Vaultier Community Edition** enables you to quickly store assorted passwords, notes and files, but a complicated installation negates its ease of use.

account to use *Vaultier*, the tool will generate a security key for you that you must store on your local disk. *Vaultier* doesn't use passwords for the user accounts, so you must point it to the key on your machine to login.

Good once it's going

The interface is fairly straightforward and there are instructions to guide you through the process of storing secrets and passwords. Information, be it passwords, notes, or files you wish to safeguard, is stored within cards. These cards reside inside vaults and the vaults themselves exist within workspaces. You can create multiple workspaces, multiple vaults within a workspace, and multiple cards within a vault.

You can add multiple secrets (passwords, files etc) to each card inside a vault. When storing passwords as secrets, you get a form which you can use to specify the website, username and password. When storing files, *Vaultier* limits the size to under 25kB.

As you continue to create more workspaces/vaults/cards to store all your secrets, locating them in a hurry may become a problem. Thankfully, the search bar at the top of the interface provides a quick means to access the stored information. It searches through all the workspaces, vaults and cards, and quickly presents all relevant matches. With the extensive

workspace/vault/card mechanism, the search is an invaluable feature.

You can collaborate with others on specific vaults or the entire workspace. In the latter case, collaborators have access to all the vaults in the workspace. *Vaultier* enables you to define the permissions for each team member. You can restrict team members to only viewing the data, or grant them permission to create vaults and cards and invite other team members.

But despite such features, and a intuitive interface with plenty of helpful hints for using it effectively, we can't really recommend *Vaultier Community Edition* for home users, because of its complicated installation process and precise requirements. **LXF**

Features at a glance

Key encryption

Vaultier doesn't use passwords and, instead, relies on a security key-based login system.

Easy search

The search trawls through the title and description fields for every workspace, vault and card.

LINUX FORMAT Verdict

Vaultier 0.7 CE

Developer: RightClick
Web: www.vaultier.org
Licence: BSD

Features	9/10
Performance	8/10
Ease of use	9/10
Documentation	5/10

» *Vaultier* must simplify its installation process or risk losing users to alternatives like *Teampass*.

Rating **7/10**

SERIOUS ABOUT HARDWARE?

10 BEST GAMING SCREENS TESTED
4K, G-SYNC, 144Hz, ULTRAWIDE OR 1080p?

MORE RAM THAN YOU CAN HANDLE
SAPPHIRE RADEON R9 290X 8GB RATED

LINUX IS NOT THE ANSWER!
HOW WINDOWS SAVED STEAM BOX

ISSUE 300/JANUARY 2015

PCFormat®

HIGH POWERED > EASY TO BUILD

£300

GAMING PC

How to build it, overclock it and play Far Cry 4 at max settings!

NOW
ON APPLE
NEWSSTAND &
GOOGLE PLAY

Download the day they go on sale in the UK!

No.1 FOR UPGRADES
GRAPHICS CARDS,
CPUs, SCREENS,
SSDs, RAM,
& MORE

Delivered direct to your door

Order online at www.myfavouritemagazines.co.uk
or find us in your nearest supermarket, newsagent or bookstore!

Hover

Add gesture and touch control to your projects, as **Russell Barnes** explores how much of the Minority Report experience this £32 dev kit will buy you.

In brief...

» Gesture and touch dev kit for your Raspberry Pi, Arduino, pcDuino or Spark Core hardware projects.

Tom Cruise first made it cool in *Minority Report* and Robert Downey Jr is still trying to top it: it seems cinema thinks there's nothing we want to do more than communicate with our computers by gesticulating manically in their general direction.

While these tantalising visions of human-computer interaction are still some way off, you can get a taste of it for just £32 (including VAT) with Hover, a tiny 6cm square development board that's compatible with a wide range of single-board computers and micro controllers, such as the Raspberry Pi and Arduino.

The premise is simple: swipe your hand up, down, left or right a few inches above the board (the website states from up to 5 inches away, but 3.5 inches was our usable limit) and the board registers your interaction. We're not talking slow and deliberate swiping motions here – a flick of the wrist in the general direction will do the job.

If you like to prod at your tech too, the board will also register touch events. It has five touch-sensitive areas: the centre and the surrounding north, east, south and west edges.

While you'll need to program the if or while statements yourself (developer Hover Labs promises updates to the library to support this more easily), the board is fully capable of registering double taps and multi-touch events. In short, an elaborate combination of hand gestures and touch events is just a sprinkling of code away.

The makers certainly deserve plaudits for making the board compatible with such a wide range of platforms. While most development boards of this ilk might just support Arduino with a rudimentary Python library thrown in for Raspberry Pi enthusiasts, Hover has full installation instructions and code examples for not one but four platforms, including the Raspberry Pi, Arduino, pcDuino and the lesser known Spark Core.

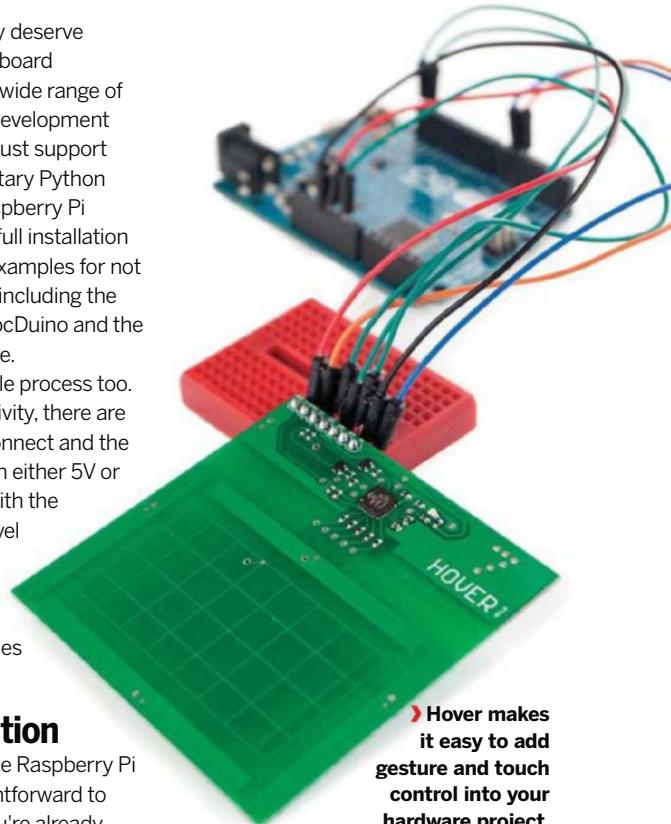
Installation is a simple process too. Besides its I2C connectivity, there are just two GPIO pins to connect and the board is compatible with either 5V or 3.3V microcontrollers with the help of onboard logic level shifting. As a rare and welcome bonus, the board's breadboard-compatible header comes pre-soldered too.

Easy configuration

We tested Hover with the Raspberry Pi and found it very straightforward to configure. Assuming you're already geared up for I2C communication, it's just a case of setting up the breadboard and downloading the provided Python library. While it's relatively basic, the library is one of the best documented we've seen for some time and it's clearly designed to help hackers and makers of all levels get the most from the hardware.

The example script for the Hover ensures you can quickly drag and drop Hover-compatible code into your project, though it would be nice for the team to update the library to support multi-touch out of the box – as it was, at least at the time of writing, the library hadn't been updated for four months.

That said, there are lots of great project examples and ideas to be found on the official Hover Labs website (www.hoverlabs.co/projects), including a section devoted to controlling retro games. But while we liked the idea of directing Frogger into oncoming traffic with a mere flick of the wrist, we were particularly taken with the video that shows a basic implementation of *Google Earth*



» Hover makes it easy to add gesture and touch control into your hardware project.

control, which uses a combination of touch and gestures.

Adding Hover to just about any computer or application is pretty easy too. Using an Arduino Leonardo or similar you can plug in a Hover as a pseudo-HID, tricking pretty much any computer into thinking it's just another keyboard or mouse. Clever stuff. LXF



Features at a glance

Easy setup

Setup via I2C is well documented on the official website for all four compatible devices.

HID-class device

Use an Arduino Leonardo to trick most systems and apps into thinking Hover is just a USB mouse.

LINUX FORMAT Verdict

Hover

Developer: Hover Labs
Web: www.hoverlabs.co
Price: £32

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

» Hover makes it incredibly easy to add gesture and touch control to just about any project you can think of.

Rating **9/10**

Football Manager

Tom Hatfield kicks off the *FM2015* season with this press conference.

In brief...

» The latest annual update to a franchise that has been around for over two decades. Core gameplay is as strong as ever, but the changes are an acquired taste.

Football Manager is often billed as 'the football RPG': selling the feeling of a manager's life as much as the strategy game built around it. It's one of those ideas that sounds fantastic on paper, but every time Sports Interactive tries to embrace the idea, it just ends up adding more press conferences.

Things taking too long is a theme of *Football Manager 2015*. Scouting players now takes several attempts, during which their attributes are displayed as a range of values that you gradually narrow down. But it's hard to

condemn the game for these irritations, because they all disappear the moment you flip over to good ol' Classic mode.

Classic values your time and gives you the tools to shortcut to the fun stuff. The user interface is a genuine improvement: it's not only pretty but also includes some smart decisions.

The bigger problem is the match engine. Criticising it is a dicey business: the way it works is so deliberately, wonderfully opaque that players tend to treat it like a force of nature. In *FM2014*, players were playing gorgeous passing football; in *FM2015*, they play like lager louts. Football d'art isn't everything, but after a few hours of *FM2015*'s muscular physicality, we found ourselves longing for the silky skills of our old team.

This might sound overwhelmingly negative, so it's important to stress

that the basics of are still there and working as well as ever. It's also entirely possible the traditional mid-season patch will change the tone of the match engine completely – at which point we'd heartily recommend it. But for now, at least, *Football Manager 2015* is a year to miss. **LXF**



» Jumpers for goalposts. The beautiful game. Men kicking a pig's bladder.

LINUX FORMAT Verdict

Football Manager 2015

Developer: Sports Interactive
Web: www.footballmanager.com
Price: £35

Gameplay	6/10
Graphics	6/10
Longevity	9/10
Value	7/10

» Slow and disappointingly retro: not a series highlight. If you're an owner of *FM2014*, it's one to miss.

Rating **7/10**

Adventures in MC

Jonni Bidwell enters the world of Minecraft for some coding fun.

In brief...

» A collection of projects using the *Minecraft* Python API, especially designed for youngsters. But also not a bad introduction to programming for everybody else.

We can't get enough of Martin O'Hanlon's *Minecraft* website (head to <http://stuffaboutcode.com>). Building on this, he and David Whale have released *Adventures in Minecraft*, a book which aims to introduce the *Minecraft* Python API to children. The text is divided into nine projects of increasing difficulty – ranging from the obligatory Hello *Minecraft* World to using Pi or Arduino GPIO-connected electronics to make a hazard-fraught

diamond-collecting game, in which you race against the (seven-segment LED) clock, which is created using

the various techniques you learned in the preceding chapters.

The text is hardware-agnostic, explaining how to set up *Minecraft* on the Pi, PC or Mac through the Bukkit server and RaspberryJuice plugin, which you can download from the book's companion website. It's also remarkably easy to follow, making it an excellent classroom resource. There are all manner of interstitial boxouts bearing extra details, explanations and warnings, as well as cartoon avatars of the authors offering their own observations. Code is kept as clean and minimal as possible, with clear explanations where necessary. Even in the grand finale project, which introduces the relatively complex matter of threading, this is broken up into easy-to-digest chunks.

With a little prodding, even the shortest attention spans will get



» Minecraft protagonist Steve is well aware of what time it is.

something out of this book. It does a great job of explaining some programming fundamentals and there are plenty of pointers to further adventures too, should young imaginations be kindled. **LXF**

LINUX FORMAT Verdict

Adventures in Minecraft

Author: Martin O'Hanlon & David Whale
Publisher: John Wiley & Sons
ISBN: 978-1-118-94691-6
Web: www.wiley.com/go/adventuresinminecraft
Price: £15
Pages: 320

» This is a great classroom/rainy weekend resource, written in an engaging style. Adult learners may find it slower-going.

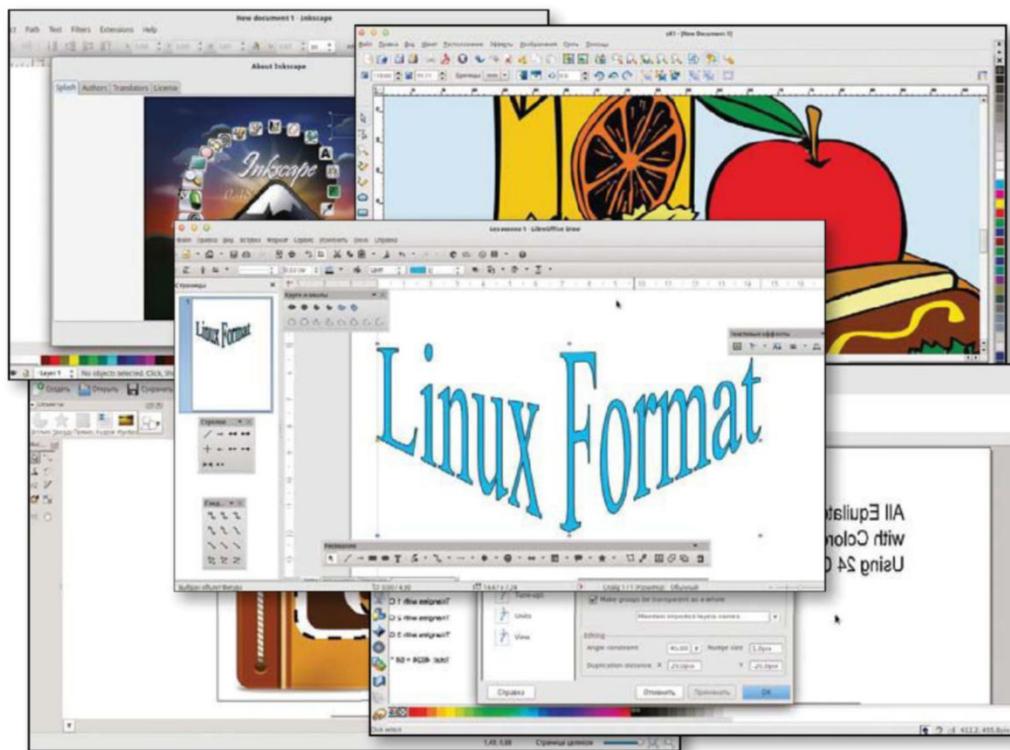
Rating **9/10**

Roundup

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

Vector editors

Looking for the best vector graphics editor to use in Linux, for illustration as well as just for charts or maps? **Alexander Tolstoy** covers all the angles.



How we tested...

All our editors were tested on the same quad-core machine with 8GB of RAM, running a recent version of the Ubuntu-based elementary OS. (Note that don't need a special distro for coping with vector graphics – you can happily use your usual preferred distribution.)

To level the playing field, special attention was paid to the 2D performance of our test machine. To minimise any system variables, we decided to use an Intel HD Graphics chip, one of a family known to offer stability, decent if not spectacular performance, and a hassle-free driver experience in Linux.

We tested one program at a time under standard conditions. We tried to select the latest stable version of each application and assessed it for usability, performance, breadth of feature set, and for the extra materials available on the web.

There was a time when vector graphics editors were widely considered special-purpose software, used mostly by professional graphic designers. These days the barrier to entry is low, which means that almost anyone can produce cool visual content, making creating vector drawings an enjoyable activity for both kids and adults.

One of the great advantages of vector graphics is scalability. Unlike raster (or bitmap) graphics, vector drawings don't get pixelated when you scale them up: you can manipulate them in any way and your drawing will

“Despite the variety of apps out there, most support the standard SVG vector graphics file format.”

always retain the initial level of detail. Despite the great variety of applications out there, most support SVG, which is the industry-standard format for scalable vector graphics.

For this Roundup we've selected five top vector editors, including perhaps the most widely known: *Inkscape* and *LibreOffice Draw* (which is part of the

popular productivity suite). The other three are *sK1*, designed for prepress tasks; *Karbon* from the KDE-centric *Calligra* office suite (formerly *KOffice*); and *Xara Xtreme*. The last of these is a bit of an oddity in the Linux world: it was released as open source software back in 2006 but implements some features of the commercial *Xara Designer Pro*.

Our selection

- » **Inkscape**
- » **Karbon**
- » **LibreOffice Draw**
- » **sK1**
- » **Xara Xtreme**

Ease of installation

What does it take to get each app running?

Much as you'd expect, the ease of installation of any given application depends on its popularity. Web browsers are readily available for almost any Linux distribution, whereas some specialised scientific applications may exist only as a source tarball.

Vector graphics tend to fall between the two extremes, meaning that if an editor is in an active state of development, such as *Inkscape*, it's widely available in any packaged format and for any Linux distribution.

LibreOffice Draw receives far less attention from developers, but being a part of a bigger package, it's generally distributed with almost every Linux distro. For those who prefer installing straight from the official *LibreOffice* site (www.libreoffice.org) rather than their package manager, there are fresh builds for 32- and 64-bit Deb and RPM systems available.

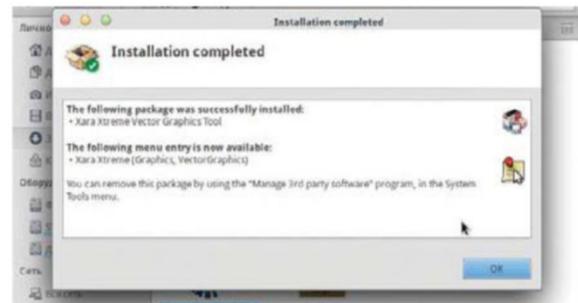
Karbon is also part of an office solution, but *Calligra* is quite a bit less

widely used than *LibreOffice*, so you might not find a pre-packaged version in some lesser-known distros.

SK1 is a very promising open source project, but it hasn't been updated for nearly two years. This isn't as much of a problem in the world of specialised graphics software with its unhurried development cycles, but *SK1* is only known to a very small group of developers and community members: even some Linux graphics enthusiasts haven't heard of the application.

As a result, *SK1* offers loads of pre-compiled packages for a dozen flavours of Linux, but frankly most of them are outdated. We managed to install packages for Ubuntu 11.04 on our 12.04 LTS based system, but we had to manually resolve some Python dependencies and use *dpkg* to install *SK1* packages. You might find it better to build it from source.

Xara Xtreme is even more elderly than *SK1* – its website looks to have been last updated in October 2008 –



► **Xara Xtreme's** Windows-like installer looks neat and tidy.

and on top of that, it comes in the fairly exotic Autopackage format. In the late 2000s this was a quite promising way to ship Linux software in a distro-independent way, but although you can try Autopackage and play with it, the chances are that something will go wrong – it doesn't really work well now after years of neglect.

Thankfully, there are enough Deb and RPM packages. You can try <http://pkgs.org> to find *xaralx* binaries for your preferred flavour of Linux. However, the application runs without any major problems once you select its latest version.

Verdict

Inkscape



Karbon



LibreOffice Draw



sk1



Xara Xtreme



► **More active development means better availability.**

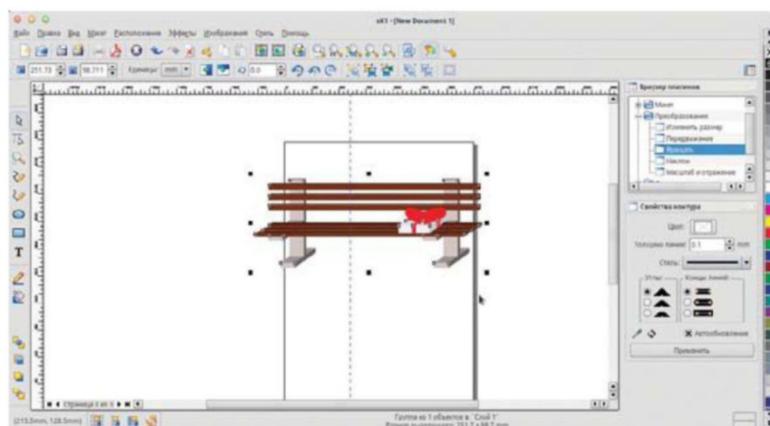
Usability

How quickly do they get you drawing?

Within this context, usability means how quickly a new user of the software can explore the features on offer and start producing drawings, or even just changing the application preferences.

Inkscape offers a very polished and streamlined interface with tool buttons

on the left, colour bars at the bottom and a tools pane on the right-hand side of the window. As soon as you need to perform custom actions, such as distributing objects or applying a set of filters, the appropriate extra panes appear and neatly stack on the right-hand side of the window.



► Despite its unfinished state, *SK1* is quite usable.

LibreOffice Draw has a main tool bar at the bottom of the window, and you can use more tools by adding extra panels from the View menu. Extra tool sets appear as floating panels, which you can position wherever you want. Being an office-centric application, *Draw* offers a huge Pages pane on the left-hand side of the window.

Karbon has an uncluttered interface with a traditional layout (tools on the left, options on the right), but many features are hard to find.

Xara Xtreme feels very welcoming, but it doesn't look tidy thanks to its outdated *wxWidgets*-based GUI, and there are also many missing icons in the editor's options.

SK1 is a very stable application, but many features are missing or display only placeholders instead of real sections, especially in the program's settings. Custom shortcuts for zooming a view or manipulating objects also take some time to learn.

Verdict

Inkscape



Karbon



LibreOffice Draw



sk1



Xara Xtreme



► **Some apps feel less finished and stable than others.**

Available tools

How many different tools and useful options are offered?

The major criterion is the variety of tools available in each editor: drawing tools, node-editing capabilities and options for dealing with images, external objects and pages, plus any unique and interesting tools to attract vector graphics aficionados. Each of

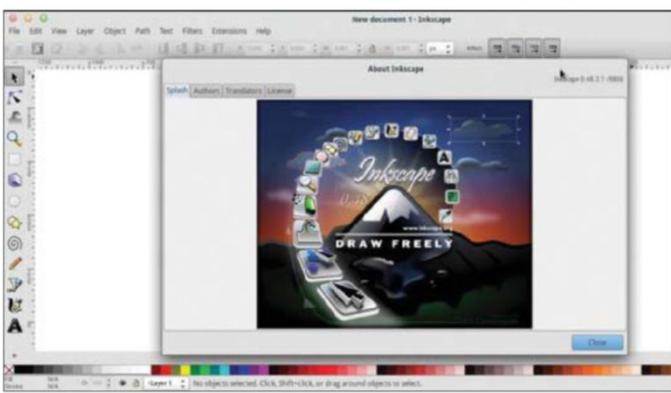
our contenders has its own specific focus, which we'll note as we go, but setting that to one side, it's clear that the broader the set of tools on offer, the better.

Of course, how feature-rich an application is, whether it's a specialist app or a general-

purpose one, usually reflects the amount of development effort behind it: after all, every toolbar button or menu item costs a lot of development time, and a stronger community or a paid development team will typically produce a more impressive product.

Inkscape ★★★★★

Inkscape is the most feature-rich and thus flexible vector graphics editor for Linux. It's actually comparable with the proprietary *Adobe Illustrator* in terms of joining and breaking nodes, manipulating object paths, and intersecting, merging and extracting objects from one other. It also offers a great variety of drawing options. There's a freehand drawing tool (a pencil) as well as a configurable brush tool, and lots of predefined shapes. It can also trace bitmap graphics and convert them into vector objects. *Inkscape* is also very good at manipulating text: it supports complex character and paragraph formatting, indents, kerning and even spell checking. Text can be wrapped around a curve or have a custom fill and outline or any combination of artistic effects applied, and still remains editable. You can also transform text into outlines at any time.



Karbon ★★★★★

Karbon is a vector companion to *Krita*. Both are part of *Calligra Suite*, and both are primarily targeted at artists. But unlike *Krita*, *Karbon*'s capabilities are very modest. We found it to be a very basic vector editor, with few advanced options. Some simple shape tools do exist, though they reside in the objects pane on the right-hand side of the window, not on the main toolbar. *Karbon* has quite a decent set of tools for manipulating nodes, lines and curves, and it can arrange and distribute objects and even use layers. There's also an 'artistic text' tool, although it proved not to be particularly artistic: just an ordinary text input tool. You can align text, add basic emphasis, and attach text to a curve, but otherwise very few formatting options are supported. Tracing bitmap images to convert them into vectors is another standard feature you won't find in *Karbon*.

Filters and effects

How far will each app take you beyond basic drawing?

After you create vector graphics, you'll very likely want to add effects to them, whether for web graphics or offline designs. Of the suites on test, *Inkscape* has by far the largest collection of effects, including dozens of shape and volume tricks, which can all be applied to an editable curve or text. There are exciting morphology, materials and texture filters for vector objects as well as a bunch of artistic filters for bitmap images. The effects sit under the Filters

menu and there's also a dedicated Filter Effects editor pane, where you can combine several effects and sort them.

LibreOffice Draw also provides a vast set of effects, though they're less artistic and more stationery-orientated. There are nice fontwork effects (cloned from Microsoft Office's WordArt), a way to wrap text around a circle, extrusion and drop shadow tools. They're not best suited to creative work, though.

Karbon has an Effects menu, but it contains only four entries for basic

outline modification. Moreover, *Karbon* can't handle SVG effects when opening a file created in another application.

sK1 is much the same in terms of effects: none to add, and no support for importing SVG effects.

Xara Xtreme frankly warns you that it's in an early state of development and thus doesn't support importing SVG files at all, but there are some interesting tools, such as bevel, mould, shadow and transparency. Better than nothing, anyway.

Verdict

Inkscape

★★★★★

Karbon

★★★★★

LibreOffice Draw

★★★★★

sK1

★★★★★

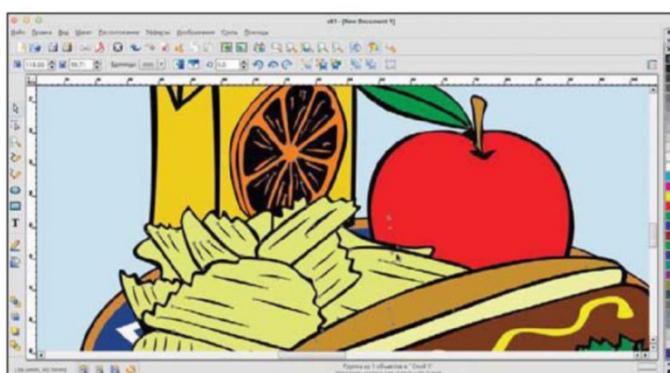
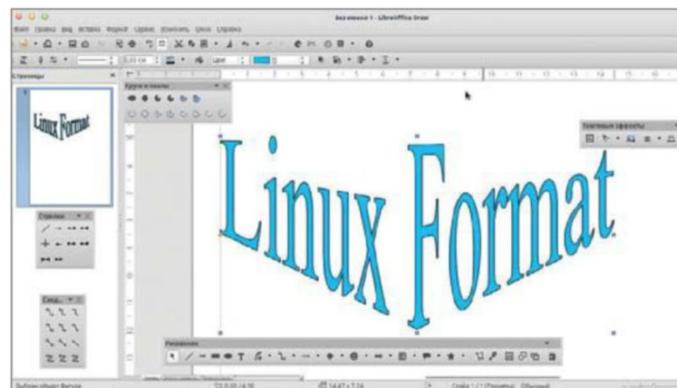
Xara Xtreme

★★★★★

» *Inkscape*'s excellent effects put it in a class of its own.

LibreOffice Draw ★★★★☆

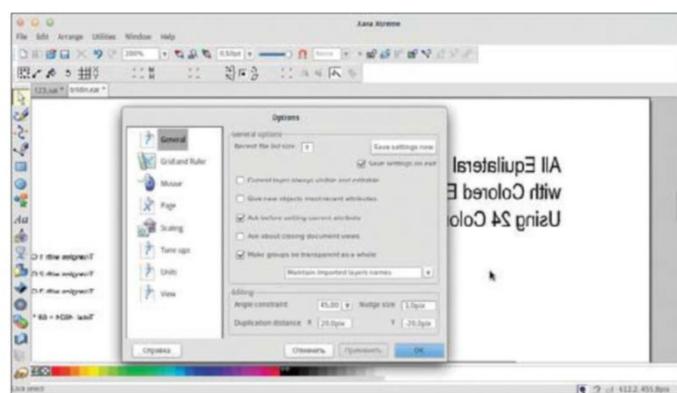
Draw has a noticeable bias towards business graphics: it's very good in designing schemes, plans, sequences and charts, and it has a very solid bitmap tracing module. There are some tools for drawing freehand shapes, which enable you to effectively draw polygonal objects (for maps, for example), but there's very little you can do with nodes and objects intersecting. It is clearly targeted at making simple shapes and organising relations between them, so you can group objects, draw beautiful linking arrows and arrange objects, but it's almost impossible – or at least very hard – to draw a photorealistic image. *Draw* is very good at simple graphics and reasonably capable for sketches and drawings, making it a great choice for business graphics, but even these could be enhanced with more artistic tools such as brushes and external effects.

**Xara Xtreme** ★★★★★

The commercial version of *Xara Xtreme* claims to have as many features as *Inkscape*, and possibly has more. *Xara Xtreme* for Linux, however, is a shadow of its commercial original. It's still quite feature-rich, though, and offers a great many node, curve, shape and blend tools, rich text formatting, colour and gradient editing, and more. Generally, it feels mature and enables you to create complex artwork made from lots of finely crafted vector objects. It ships with a nice set of vector clip art and also has a built-in bitmap tracer. Many features are left unimplemented – such as advanced freehand and brush tools, or a bitmap editor – but *Xara Xtreme* still makes a good companion for digital artists. It offers cool bevel effects and adjustable shadows, as well as precise control over lines and curves, plus kerning and tracking support for text.

sk1 ★★★★☆

SK1 quite successfully recreates some key features of professional DTP software in an open source project. It includes many publishing features, such as CMYK colour space support, separations, ICC colour management and PDF output. The left-hand part of the window contains a plugin browser with very nice tree view, which makes *sk1* look like a mixture of DTP application *Scribus*, and *Skencil*, its precursor. There are some drawing tools, but they are very basic, as are the options for working with nodes and curves. *SK1* is more suitable for manipulating objects and preparing graphics for printing, rather than for artistic or creative uses. It also feels unfinished, with so many features missing, which is a great shame.



Format compatibility

How good is support for industry-standard vector formats?

In publishing and vector graphics there are a few core file formats that are essential for reliable data storage and file interchange. You need Portable Document Format (PDF) and Encapsulated PostScript (EPS) for saving preprint page layouts, Scalable Vector Graphics (SVG) for storing curves and outlines, and the proprietary AI and CDR formats for handling *Adobe Illustrator* and *CorelDRAW* files.

PDF and SVG formats look like the must-haves. Except for *sk1*, each editor

has PDF import support, but *Karbon* failed to parse our test PDF file and *Xara Xtreme* imported it as a bitmap picture (everything was merged and rasterised). *LibreOffice Draw* has a very decent PDF importer and exporter.

SVG support is perfect in *Inkscape* and quite good in *LibreOffice Draw*, *Karbon* and *sk1* (although the latter three support only basic SVG features). *Xara Xtreme* doesn't have any support for SVG and generally has rather weak format support.

Some time ago, the *sk1* project produced *UniConvertor*, a tool that can read many proprietary formats, including AI up to v9, EPS, *CorelDRAW* X3 and X4, and more (see <http://bit.ly/UniConvertor>). *UniConvertor* is the best open source solution for reading these formats. It's used by default in the *sk1* editor and is automatically detected by *Inkscape*. *Xara Xtreme* has its own implementation of AI and EPS support and can securely write in these formats, though reading is less reliable.

Verdict

Inkscape
★★★★★
Karbon
★★★★★
LibreOffice Draw
★★★★★
sk1
★★★★★
Xara Xtreme
★★★★★

» *SK1* could have scored better, but PDF support is poor.

Performance

How well does each app cope with the heavy lifting of vector files?

Unlike bitmap graphics, even very complex vector graphics are small files, so handling them is a test of an editor's processing capabilities, not your hard drive read speed. Our sample SVG file, despite being only 2MB in size, contained 18,496 objects, most of them bordering

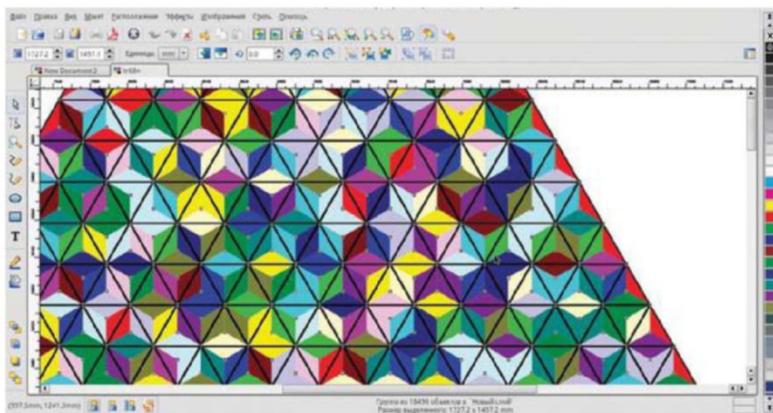
each other. Our test involved trying to open the file and then mirror all its objects. Processing the file was a tricky task for some of our editors, but the results were surprising. *Inkscape* didn't acquit itself very well: it was slow at zooming complex SVG outlines and generally was very uncomfortable to

use. It opened the test file only after a 40-second delay, and any manipulation involved significant pauses.

LibreOffice Draw and *Karbon* were both extremely slow in opening the test file and performing any modification of the objects. It took several minutes to complete a few simple actions, and we wouldn't want to use these applications for heavy vector graphics.

With *sK1*, the story was entirely different. Though *sK1* uses the same Cairo renderer found in *Inkscape*, the performance and responsiveness were many times faster and we could comfortably play with our test file.

Xara Xtreme couldn't import SVG, so we converted the test file into PDF in *Inkscape*. *Xara Xtreme* relies on its own renderer, which is known to be very fast and well-optimised. This was the case with our endless small triangles. Selecting and mirroring over 18,000 objects took only a few seconds – a mind-blowing performance.



► This garish composition of more than 18,000 vector triangles finally brought most of our editors to their knees. But some still survived the test.

Verdict

Inkscape

★★★★★

Karbon

★★★★★

LibreOffice Draw

★★★★★

sK1

★★★★★

Xara Xtreme

★★★★★

» *Xara Xtreme* and *sK1* display astonishing speed here.

Documentation and support

How much useful information do you get or find if you search online?

Finding answers to your questions is very important and often the reason why you're likely to stick with certain pieces of software. Let's see how well documented our vector graphics editors are.

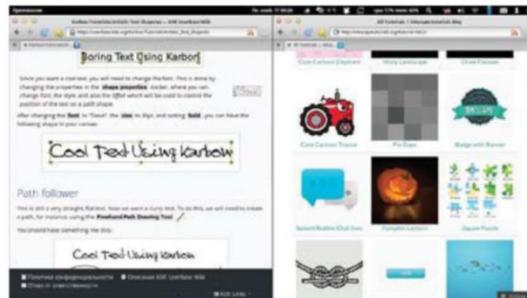
Inkscape has a great online manual, a wiki at <http://wiki.inkscape.org> and a clutch of both official and third-party tutorials. Even if you've never used *Inkscape* before, you can fall in love with its gorgeous sample graphics and learn how to recreate them step by step. Searching the web for particular answers also turns up a good range of forums and threads dedicated to this premier graphics editor.

LibreOffice Draw has a nice but small section in the main *LibreOffice* portal at <http://help.libreoffice.org>. There are guides and tutorials for basic drawing tools and for working with objects, some of which reside at third-party tutorial portals. The amount of documentation is adequate for the software's modest feature set.

Karbon has only a small page inside the general documentation for *KOffice* (the former name of *Calligra Suite*). There's one tutorial on working with artistic text shapes in *Karbon* and that's it. Searching the web for anything more largely led us to completely unrelated sites about fuel and heating...

sK1 hasn't shown much activity in recent years, but there is a nice support forum for both *sK1* and *UniConvertor* (<http://sk1project.org/forum>), with answers to common questions. A quick search also reveals a fair number of community reviews and introductory posts about installation and features, which can be very helpful. Considering *sK1*'s modest feature set, we'd say that no major issues remain unanswered.

Searching for *Xara Xtreme* materials results in plenty of links related to Xara's commercial tools. But if you look at the <http://xaraxtreme.org> website, you'll find tons of carefully selected documentation. The site hosts very detailed and helpful FAQs and links to the program's archived forum page (<http://bit.ly/XaraForum>). Although the editor itself has been left in an unfinished state, its documentation is mature and looks professional. Again, it's appropriate for the feature set.



► *Karbon's lonely manual can't compete with comprehensive library available for *Inkscape*.*

Verdict

Inkscape

★★★★★

Karbon

★★★★★

LibreOffice Draw

★★★★★

sK1

★★★★★

Xara Xtreme

★★★★★

» *The amount of support is in line with each app's feature set.*

Vector graphics editors

The verdict

The last time we covered vector graphics editors in a Roundup was way back in **LXF79**, but the winner this time is the same as in 2006. *Inkscape* does an impressive job in handling most aspects of vector graphics. It offers an uncluttered and intuitive interface, a great tool set for drawing curves, and an almost endless number of eye-catching filters and effects. For years *Inkscape* has been evolving into a 'nearly professional' editing application, capable of both artistic uses and basic DTP tasks. Thanks to *UniConvertor*, it is able to import many (if not all) commercial-grade AI and CDR format variations – and if you run into trouble, lots of support and topic threads with solutions can easily be found online.

Xara Xtreme is not as complete, but it deserves second place thanks to its fast renderer, decent format support (especially for exporting) and good documentation. It's hard to predict

whether someone will ever pick up the software development, but *Xara Xtreme* remains usable, fast and free.

LibreOffice Draw does its job well enough; it's just not meant for artistic purposes and offers only a relatively modest set of tools. That said, you'll find it very capable of producing business graphics, schematics and simple drawings. It's stable and solid, and has the advantage of being part of the *LibreOffice* suite, which to a degree compensates for its own limitations.

In fourth place is *sK1*, a very DTP-centric solution with some handy vector drawing tools, a fast and responsive renderer and the fantastic *UniConvertor* utility. There is an attempt to develop a successor to *sK1* called *PrintDesign*, but activity seems slow

– the last preview was released in October 2013 and that was for Ubuntu 13.04. Surely *sK1*



» You can find tutorials about creating such showy artwork (using *Inkscape*, in this case) on the web.

deserves closer attention from new users and developers?

Finally, although it's aimed at artists, *Karbon* seems to have been left behind by the march of time. It had more supported features in 2006 than it does now, and it currently stands no chance of becoming an *Adobe Illustrator* killer for the Linux world.

"Inkscape is impressive in most respects, and suitable for both artistic uses and basic DTP."

1st *Inkscape* ★★★★☆

Web: <http://inkscape.org> **Licence:** GNU GPL **Version:** 0.48.5

» A pleasant-to-use and feature-rich premier vector editing application.

4th *sK1* ★★★★☆

Web: <http://sk1project.org> **Licence:** GNU LGPL **Version:** 0.9.1

» Fast and quite simple to use, but not really feature-rich.

2nd *Xara Xtreme* ★★★★☆

Web: www.xaraxtreme.org **Licence:** GNU GPL **Version:** 0.7

» An unfinished but very promising professional vector graphics editor.

5th *Karbon* ★★★★☆

Web: www.calligra.org/karbon **Licence:** GNU LGPL **Version:** 2.8.6

» The curse of *KOffice* still hangs over this tiny application.

3rd *LibreOffice Draw* ★★★★☆

Web: www.libreoffice.org **Licence:** GNU LGPL/MPL **Version:** 4.3.4

» A solid and tightly built editor to fulfil your office vector graphics needs.

Over to you...

Do you use great vector graphics software that you want to tell the world about? Let us know at lxr.letters@futurenet.com

Also consider...

For our Roundup, we selected the five most prominent vector graphics editors for Linux. But there are more applications to explore. It's well worth having a look at *Sketsa SVG Editor* (\$89, www.kiyut.com), the SCADA-inspired *GLIPS Graffiti* editor (<http://glipssvgeditor.sourceforge.net>) or *Xfig* (www.xfig.org): old-fashioned, but still active and usable.

Even long-abandoned bits of software such as *Sodipodi* are sometimes worth looking at, if only for educational purposes.

You might have noticed that most of the editors here are cross-platform, but you can go even further and choose an online vector editor. Check out *Janvas* (www.janvas.com): a Chrome/Chromium extension primarily meant

for web graphics production. There are many others, including Google's SVG-edit and the open source Vector Magic (www.vectormagic.com). Most such editors are easy to use and learn, and also provide help for newcomers. There's an abundance of choices, including great bits of open source software for any taste, welcoming you to contribute. **LXF**

Subscribe to

LINUX
FORMAT
Get into Linux today!

Subscribe to our print edition, digital edition or get the best value with our complete print and digital bundle!

Choose your **LINUX** package

Print



Every issue delivered to your door with a 4GB DVD packed full of the hottest distros, app, games, and more.
PLUS exclusive access to the *Linux Format* subscribers-only area.

ONLY £25.49

Your subscription will then continue at £25.49 every 6 months – SAVING 40% on the shop price.

Digital



Instant access to the digital editions of the magazine on your iPad, iPhone and Android devices.

PLUS exclusive access to the *Linux Format* subscribers-only area, featuring complete issues & disc downloads.

ONLY £20.49

Your subscription will then continue at £20.49 every 6 months – SAVING up to 37% on the shop price.

Get the complete **LINUX** package

LINUX
FORMAT
Get into Linux today!

Print + Digital



- » A DVD packed with the best new distros and free & open source software every issue.
- » Exclusive access to the *Linux Format* archive – with 1,000s of DRM-free tutorials, features, and reviews.
- » Every new issue of the magazine in print and on iPad, iPhone, and Android devices.
- » Never miss an issue, with delivery to your door and straight to your device.
- » Huge savings, the best value for money, and a money-back guarantee.

**ONLY
£30.99**

Your subscription will then continue at £30.99 every 6 months – SAVING 40% on the shop price and giving you up to a 83% discount on a digital subscription.



Two easy ways to subscribe...

Online: myfavouritemagazines.co.uk/LINsubs
Or call **0844 848 2852**
(please quote PRINT14, DIGITAL14, BUNDLE14)

Prices and savings quoted are compared to buying full-priced UK print and digital issues. You will receive 13 issues in a year. If you are dissatisfied in any way you can write to us or call us to cancel your subscription at any time and we will refund you for all unmailed issues. Prices correct at point of print and subject to change. For full terms and conditions please visit: myfavm.ag/magterms. Offer ends 22/01/2015

BUILD A LINUX MEDIA SERVER

Turn your PC into a hub to stream all of your movies, TV shows, music and photos to any device that you choose. Nick Peers hands you the remote.



There's one compelling reason to build a media server: the ability to access all of your media on tap. A media server puts you in control, saving you the hassle of fiddling about with DVDs or CDs as well as the inconvenience of tying yourself to a single spot in your home in order to watch a movie, share a photo or listen to some music.

All you need – apart from your media of course – is a suitable Linux-powered machine to act as your server. Whether you're treating yourself to a brand spanking new machine or want to give an older, but perfectly serviceable, desktop or laptop a new lease of life, we've got you covered. Your biggest worry is choosing which

media server to run. We're focusing on *Kodi*, previously known as *XBMC*, in this feature and revealing add-ons that can help fill in some of its feature gaps, as well as pointing you to alternatives if you're willing to sacrifice the FOSS principle for more functionality and

Kodi doesn't (yet) support photo sharing, and look at ways of streaming music to AirPlay speakers and other wireless devices.

With your media server set up and ready to serve, all that remains is to work out how to access your media from other devices.

In the final section of our feature, we'll go through your various options, revealing the apps, tips and tricks required to let you watch or listen to your media on a variety of devices, from another PC or

mobile (Android and iOS) to your Smart TV, set-top box, dedicated Raspberry Pi media centre or streaming device, including Google's Chromecast and the Apple TV. Ready to transform the way you consume media in your home? Then read on.

| “We’ll reveal everything you need to know about setting up your media libraries.”

integrated functions. We'll then reveal everything you need to know about setting up your media libraries, from naming and organising your files to adding them to your server. We'll look at ways in which you can integrate your photo libraries even though



Choose your server

Get the right hardware and software for your media server.

The first question to ask yourself is how you want your media served. On the hardware front this boils down to two basic choices: you can press your existing desktop or laptop into service, or you can opt for a dedicated device whose sole purpose will be to serve media to wherever and whoever demands it. The advantages of using your existing PC are obvious: cost and minimal setup. Just install your media server on your PC and jump straight ahead to p36.

However, you do have to ask yourself how much of an impact will an always-on media server have on your day-to-day computing. If you simply want something that runs in the background performing no other tasks, then a dedicated PC is the way to go. It needn't cost you any money either, if you have an older laptop or desktop lying around. Just make sure it has enough grunt for the task at hand.

You'll need a 2GHz dual-core processor coupled with a graphics chipset that supports hardware-accelerated video decoding (Nvidia GeForce 8 series or later; Intel's GMA X4500HD or later; or AMD Radeon R700 (HD 4000) or later), which basically covers pretty much any PCI-e graphics card. In terms of

memory, 4GB RAM is more than ample, but consider upgrading the boot media to a cheap solid state drive, fast SD card or USB flash drive (32GB is sufficient, or even just 4GB if you choose to use OpenELEC) with your media stored separately, either on a second internal drive, USB drive or even a NAS drive.

If you're looking to buy a new PC, then the Raspberry Pi looks a cheap bet, but it's better suited as a media player rather than a server. Instead, take a look at a nettop PC from the likes of Zotac or Acer, or build your own mini-PC using a small form factor Mini-ITX-based board and a suitable case, such as Antec's ISK 110 mini case with VESA mount.

The right software

If your server is performing double duty as a media centre under the main or bedroom TV, a wireless keyboard and mouse is obviously a good idea for convenience's sake. Also consider going down the HomePlug route if your server isn't directly connected to your router, particularly if you plan to stream HD files (pick a 500Mbps or 600Mbps model to ensure stutter-free streaming).

There are a multitude of choices when it comes to picking the software that will

organise and stream your music, movies and photos to all your other devices. We've picked *Kodi* – better known in its previous incarnation as *XBMC*. Even though it's geared more towards being a media centre, it's a capable of being a media server too, and comes with the main advantage of being free and open source, with no strings attached. *Kodi* can be installed on to an existing Linux installation, but if you're building a dedicated media server consider choosing one of the versions bundled as a standalone OS.

Choose XBMCbuntu (<http://kodi.wiki/view/XBMCbuntu>) if you'd like access to an underlying Linux installation (based on Ubuntu LTS), or OpenELEC (<http://openelec.tv>) if you're running a lesser-powered machine and simply want access to *Kodi* without worrying about any additional configuration.

Whether you plan to install *Kodi* on your existing Linux machine or decide to build a dedicated media server, the installation is straightforward – links to the appropriate repositories or source code, along with instructions for installing, can be found at the *Kodi* wiki (<http://bit.ly/XMBC4Linux>). Once it's installed, you're ready to start setting up your server.

Alternative media servers

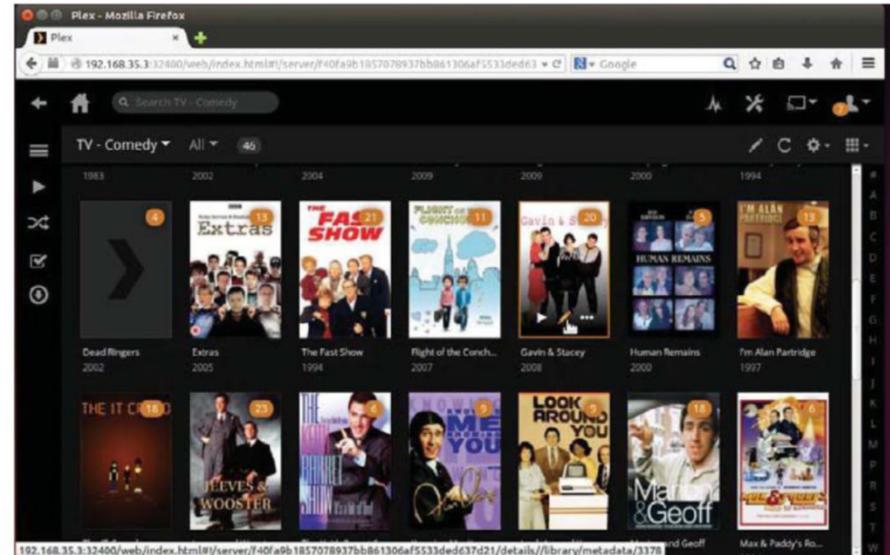
Although we're focusing on *Kodi* in this feature, you'll find a rich choice of alternative media servers out there, some of which may be more suited to your personal needs. The most visible alternative is *Plex Media Server* (<https://plex.tv>) – which, if we're honest, is a more powerful and versatile media server solution, and may appeal to those who want a reasonably straightforward setup procedure and don't want to spend time sourcing workarounds for some of *Kodi*'s missing features. However, it isn't open source and charges for premium features as well as for its mobile apps. It is universally supported, though, and you can install *Plex Media Server* alongside *Kodi* on your server and tie the two together with the free *PlexBMC* add-in.

In a similar vein, *Media Browser* (<http://mediabrowser.tv>) is making a concerted tilt for *Plex*'s crown, and is now available for both Ubuntu and Debian (in beta). It's similar to *Plex* in that it provides a polished interface for a wide range of media, and supports a number of living room and mobile devices too, but again charges for certain apps.

If you're trying to install a media server directly on to a Linux-embedded NAS drive, then *MediaTomb* (<http://mediatomb.cc>) or *Serviio* (<http://serviio.org>) are worth considering.

Both serve to any Universal Play and Play (UPnP) compatible device, but if you want to access non-DLNA (Digital Living Network Alliance) devices in *Serviio* you'll have to purchase a Pro licence.

If you're primarily into streaming music, *Subsonic* (<http://subsonic.org>) is worth considering. This is a web-based server that enables you to access a number of supported devices both inside and outside your network.



► **Plex Media Server** is a well-known alternative to *Kodi*, but charges for premium features.

Set up and configuration

The hardest part of the process is organising and configuring your media libraries – here's what you need to do.



Once *Kodi* is installed – either on its own or as part of OpenELEC or XBMCbuntu – the first thing you'll need to do is get your media organised. You can physically store your media on any drive accessible to your server, but we recommend locally connected drives, if possible, as streaming content stored on a NAS drive could be problematic, particularly when accessing HD video.

Wherever you choose to store your media, your first task is to organise your media accordingly. That means setting up a logical folder structure into which all your media is stored, and naming your media files correctly (see below). The advantage of doing this is that it enables *Kodi* to scan and identify files, pulling in metadata and artwork from the internet to deliver a media library that's packed with useful – and identifiable – information that makes it easy to browse from other devices.

You'll find detailed instructions for organising your files at the *Kodi* wiki (<http://bit.ly/XMBCNamingFiles>). For instance, to enable proper scanning, movie film file names should only include the title and the year (optional) followed by the file extension. Thankfully, renaming your files doesn't have to be a chore thanks to *FileBot* (see p37) which can also add in useful additional information, such as the episode titles to each individual show to make the files more recognisable. The step-by-step guide (p37) reveals how to use it to rename your TV media, but it can also be used for movies and music.

Once your media is correctly named and stored in the right location, it's time to set up *Kodi* itself. Launch it if necessary to find yourself at the main screen. If any add-ons are flagged as broken at this point, click Yes to remove them. Next, navigate to the System section and choose Settings to set up your server's core functionality. Start by selecting Services > UPnP and click Share video and music libraries through UPnP to turn *Kodi* into a bona fide media server. Then work your way through the rest of the section enabling what features you think you might need – switch on

the Webserver settings for example, and you'll be able to administer remotely from another computer by typing its IP address into your browser using the following format:

192.168.x.y:8080. Make sure you password-protect this feature.

Also consider changing the skin if you don't like *Kodi*'s look. Switch to the Appearance > Skin section under Settings and choose something like PM3.HD, if you'd rather menus were displayed vertically rather than horizontally, for example. Most other settings affect *Kodi* as a media centre rather than a media server (see p38 for details).

Put your library together

The moment of truth is here: *Kodi* is set up and ready to go, and so are your media. Now is the time to link them up. This involves creating libraries for different media types – you could simply have basic libraries like Movies and TV Shows, but if you have a large collection you may prefer to split them into genres – Movies (Sci-fi) and TV (US comedies) etc – to make them easier to browse from any device.

However you've carved up your media, the procedure for adding each library is the same: return to the home screen and select Videos > Files and you'll see a solitary option awaits you: Add Videos. Click this to add your first library – or 'source' as *Kodi* calls it. Each library points to one or more folders. Assuming you've organised your libraries accordingly, you'll only need to direct it to one folder – click Browse to do so. Next, give your library a suitably descriptive name (TV – Comedy, for example) and then you'll be prompted to Set content, which tells *Kodi* what kind of media to expect. First choose its type (Movies, TV, Music Videos or Music), pick a scraper (the content database you want *Kodi* to use for artwork and media information) and then tweak your preferences.

Once complete, select Yes when prompted to refresh the library and wait while *Kodi* scans the folder and matches up its content with the selected scraper. Once you've set up your first library, you can move on to the next one – navigate back to the top of the Videos > Files >

Add Videos to set up your new source. We'll cover how to add music to your library in more detail later (see p38), and reveal workarounds that enable you to share photos too.

Extending Kodi

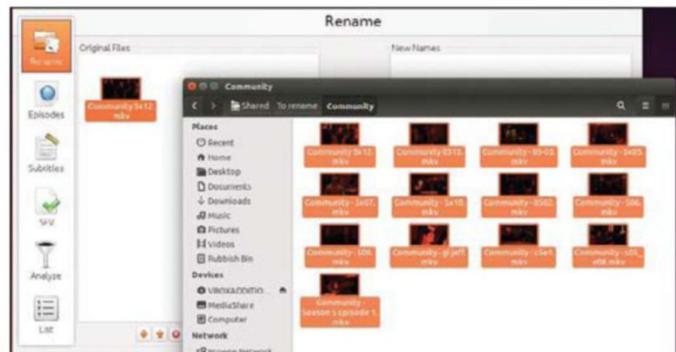
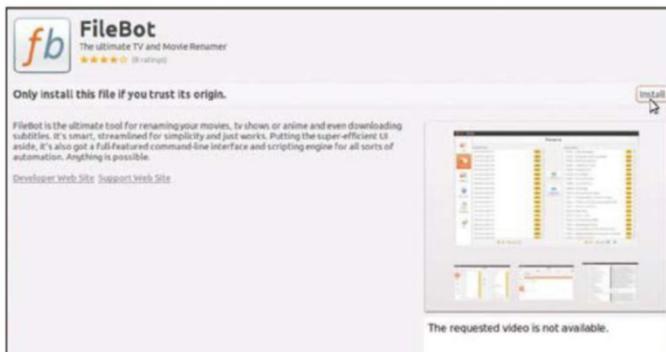
Kodi's is a perfectly adequate media server if your content is readable on all devices and you only want to access it over your local network. If you want remote access over the internet to your media, or you have devices like Chromecast that refuse to recognise some or all of your media due to the video format, you'll need some extra help. Because *Kodi* can't transcode media like alternatives, such as *Plex* or *Media Browser*, you need to enlist some further help from *BubbleUPnP Server* (<http://bit.ly/BubbleUPnPServer>).

BubbleUPnP is designed to sit alongside an existing server and provide missing functionality. It offers the ability to transcode media and provide access to it from outside your network, via your Android tablet or phone, which can then pipe it to other devices. The Android app is free to install, but costs £3.04 to unlock certain limitations, so try before you buy. *BubbleUPnP* is reasonably simple to set up – full instructions can be found at the website. Once done, open your server's browser and type **localhost:5050** to configure it. So long as *Kodi* is running in the background you should see it immediately under the Media Servers tab.

If you've got access to a USB TV tuner device, why not get your server to perform as a video recorder in addition to its media server duties? This will enable you to record your favourite programmes and then watch them on any device you choose at a later date. *Kodi* works in conjunction with a number of TV back-ends, including *TVheadend*, *VDR*, *MythTV* and *DVBLINK*. The best known of these is – of course – the open source *MythTV PVR* project, which can be found in most repositories, and promises to work with any tuner that's supported in Linux. (We'll be taking a more in-depth look at how to add PVR capabilities to your Linux PC in a future issue.)

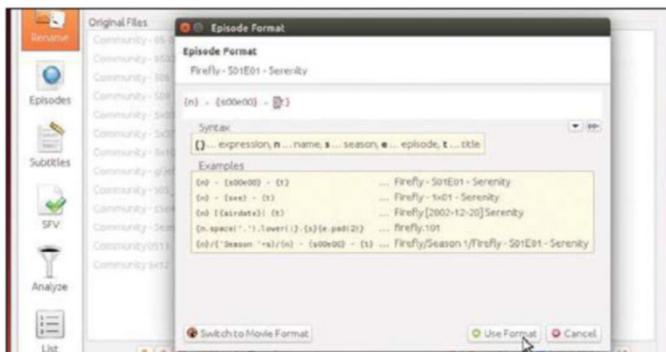
Type	Folder Structure	Syntax	Example
Music	Music\Artist\Album	artist – track name	Music\The Beatles\Past Masters Volume 2\the beatles – paperback writer.mp3
Movies	Movies\Genre\Movie Title	title (year)	Movies\Sci-Fi\Star Trek\star trek (2009).mkv
TV shows	TV\Genre>Show Title\Season	tvshow – s01e01	TV\US Comedy\Community\Season 3\community - s03e04.mkv
Music videos	Music Videos\Artist	artist – track name	Music Videos\A-ha\A-ha – did anyone approach you.mkv

Rename media files quickly with FileBot



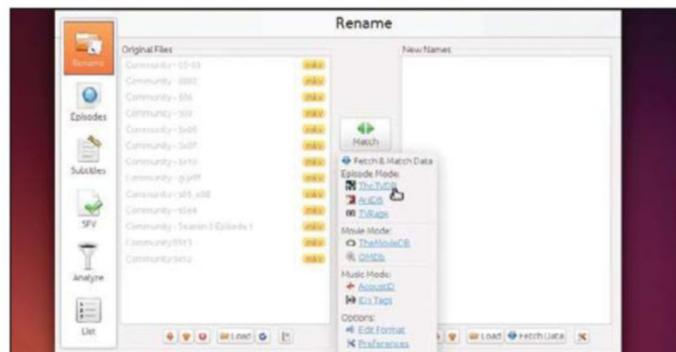
1 Get FileBot

You'll need Java 8 or later installed in order to use *FileBot*. Once that's in place, download the appropriate Deb package from <http://filebot.net> and install it by double-clicking the file and following the prompts (Ubuntu users can – for a \$4.99 donation – also install it directly from the *Ubuntu Software Center*). Once installed, launch it via a search for *FileBot*, or by typing **filebot** into a Terminal window.



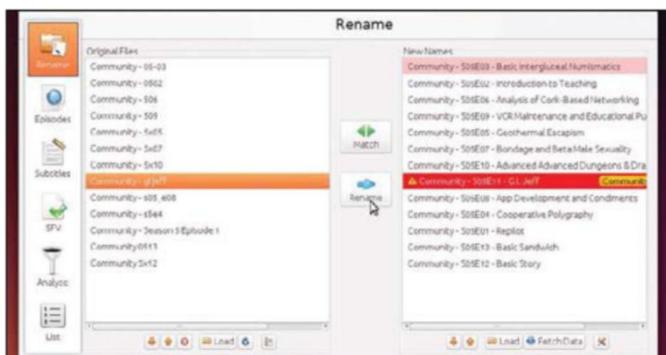
2 Select episodes

Open a file window and browse to the folder containing your first TV show. We recommend renaming in batches to make the process more manageable, so either drag the show's folder on to the *FileBot* window to select all of its episodes, or open up the folder and drag one or two season folders into the window. You'll see the episodes you've selected appear, greyed out, under Original Files.



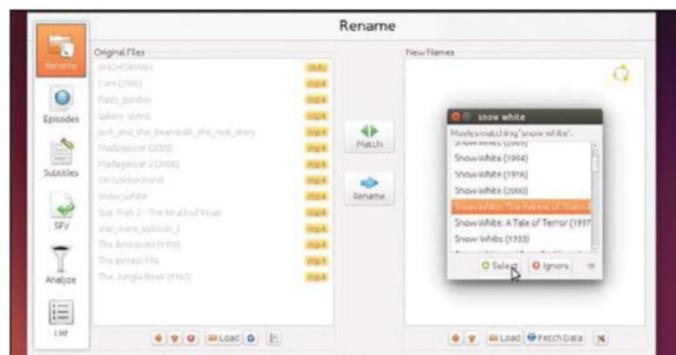
3 Choose naming format

Next, click the Match button and choose Edit Format to set your naming convention for TV episodes. You'll see a number of examples – click one to select it or use the (x)= button to reveal more common 'bindings', such as airdate or genre. To stick to our suggested format (see p36), plus include the episode title to help identify the file, type **{n} - {s00e00} - {t}** into the box and click Use Format.



4 Match episodes to database

Click the Match button again, but this time select one of the three supplied Episode Mode databases – typically the TheTVDB is the best choice, so start with that. It'll search for matching show titles – if it finds more than one possible match you'll need to select which one you think is correct before clicking Choose. It'll then display a list of suggested new filenames in the New Names box.



5 Review and rename

Including the episode titles should help you quickly determine if the match is genuine or not – if not, try another database or pick a different suggestion from those provided. Pay particular attention to matches marked red or pink; these *FileBot* isn't so sure about. If you're happy with them all, click Rename; remove any incorrect matches first by selecting the episode and pressing the Delete key.

6 Move on

If any titles contain illegal characters that can't be used in filenames, *FileBot* will throw up a warning. You only need to click Validate to remove the characters. Once complete, you can move on to the next batch of episodes. Alternatively you can rename movies or music by clicking the Edit Format button, choosing Switch to Movie Format or Switch to Music Format to set their naming format.



Stream music and photos

Access and share photos and music from *Kodi* and other sources.

Your media library is obviously split into three distinct types of media: video, music and photos. You may have already added your music library to *Kodi* following the advice on the previous spread. It's simple and easy to do with the right naming convention and folder structure, and *Kodi* supports a wide range of formats from the ubiquitous MP3 to other widely supported formats, from WMA to AAC, and FLAC for the purist.

Adding music to your library is the same process as for video: select the source, give it a name and then let *Kodi* scan the contents into your library. If it misses this final step you'll find you can only browse your music library as a series of folders, and Library view will be greyed out as an option when you click the menu sidebar. If this happens, right-click the folder you just added and choose Scan item to library – you should see the scan progress bar appear and once it's complete Library view will be available, enabling you to browse your library by genre, artist and other criteria.

Once added to your library, your music will be accessible to you through other devices in the same way as your other media (see p40 for details of how to do this from different types of device). You can also output your music to

various different wireless speaker systems. Bluetooth users should take a look at *Blueman* (<http://launchpad.net/blueman>), for example. AirPlay speaker owners can also – indirectly – stream music directly from their server, but not through *Kodi* itself, sadly.

Stream music over AirPlay

If you'd like to be able to stream music to AirPlay-enabled speakers from your server without having to resort to a third-party device such as your mobile, then you may find the solution lies in the *PulseAudio* sound system found on most Linux distros (but not, crucially, on XBMCbuntu). While AirPlay won't work in this way with *Kodi*, it should work with other media players, including *VLC*, *Amarok* and Gnome's *Videos* and *Rhythmbox*.

If you're outputting to an older AirPlay speaker, and your Linux distro has *PulseAudio* installed, then you may already have all the tools you need to hand: from the Terminal type **paprefs &** and tick the option Make discoverable Apple AirTunes sound devices available locally. Click Close, open the Sound Settings dialog and your AirPlay device should be listed as an output device. Select this and then test it using *Kodi* or another music player to see if it works.

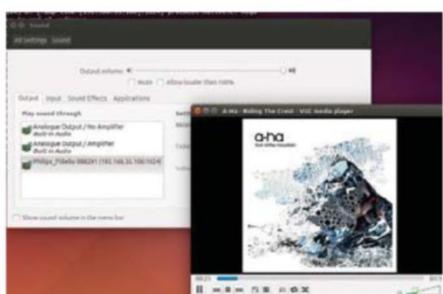
If this fails, or you have a newer AirPlay device – including the Apple TV 3 – then try the *PulseAudio RAOP2* project (<http://bit.ly/raop2>) where you'll find instructions written for Ubuntu 14.04, but which are transferable to other Debian-based distros. Follow these to install the prerequisites, retrieve the code and build the software. You then need to tweak *PulseAudio* preferences, kill the existing *PulseAudio* process, then copy a configuration file before relaunching the new process. We tested it successfully with our Philips Fidelio, but beware a lengthy lag.

If you want to stream music from *Kodi* to your AirPlay speaker, you can use your smartphone as an intermediary. When looking for an app, be aware that some freebies (such as *Strembels* – <http://strembels.com>) will play some formats – typically MP3 – but not others (AAC or WMA) so test them thoroughly first. If you're happy to pay for the privilege of streaming to your speaker, Android users will find *AOA Player Airplay* (<http://bit.ly/AOAPlayerAirplay> – £2.17) will work with most common formats, as will *8player* (<http://bit.ly/8Player> – £2.99) on iOS.

Sharing photos

The DLNA and UPnP standards support the sharing of photos as well as streaming of music and video across your network. Bizarrely, *Kodi*'s UPnP support encompasses video and music, but not photos, so you're going to have to explore other avenues. You could, of course, switch wholesale to *Plex Media Server* or *Media Browser* (both support photo sharing), but you might also want to consider running *Plex* alongside *Kodi* – if you do, install the *PlexXBMC* add-in to give you access to your *Plex*-hosted photos through *Kodi*. Download this as a ZIP file (<http://bit.ly/PlexXBMCAddIn>) and install it via System > Settings > Add-ons by returning to the top level and selecting Install from zip file. The step-by-step guide (see p39) reveals how to add photos to your *Plex* library, which can then be accessed through any of the clients referred to over the page.

Chances are most – if not all – of your photos are stored online, in which case you may already be accessing them on a variety of devices using whatever applications are available. Don't assume these applications will be restricted to your desktop or mobile devices either; you'll find most popular services offer



► It's a little inelegant, but you can stream music from your server to AirPlay speakers.

Use Kodi as a media centre

Kodi's primary function is as a media centre, but why not let it perform double duty by connecting it to a suitable TV? To use *Kodi* in this way, you should explore its many add-ons, which enable capabilities such as catch-up TV. The complete list of 'official' plugins is at <http://addons.xbmc.org>, but there are unofficial repositories, such as <http://superrepo.org>, that offer over 1,300, including ITV Player. To add a new repo to *Kodi*, you must download the repo's own ZIP file via

your browser. Once done, switch to *Kodi*, browse to the root of the Apps section under System > Settings and choose Install from zip file. Once installed, click Get Add-ons and you'll see the repo is now an additional option.

You can also download individual add-ons directly as ZIP files too – a good list can be found at the *Kodi* wiki (<http://bit.ly/KodiAddOns>), which is where you'll want to go to install the BBC iPlayer add-on, for example.



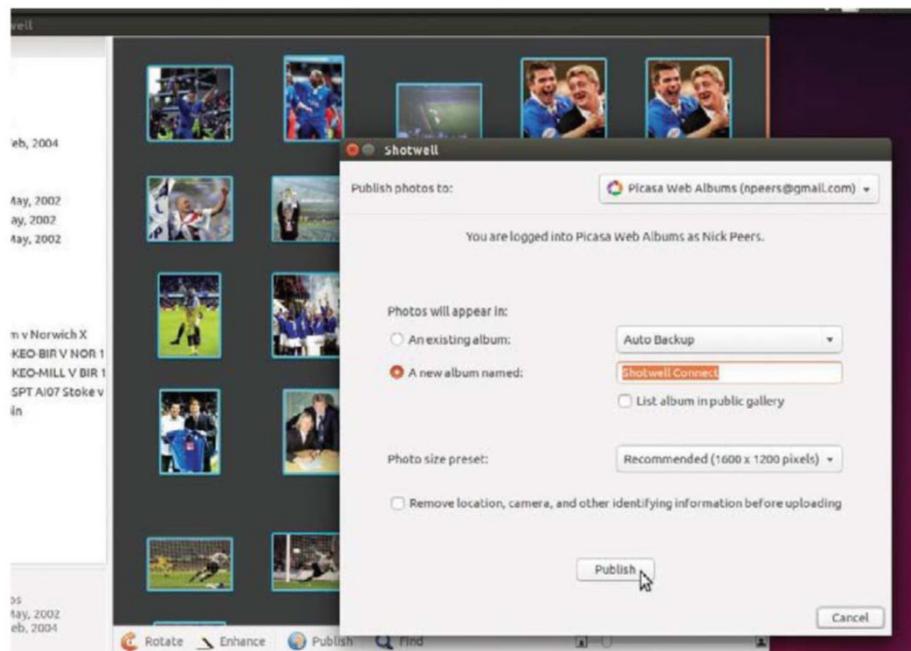
apps on other platforms too, such as your smart TV or Blu-ray player. When setting up these apps, also look for a Pictures add-on in *Kodi* to see if you can then view your photos when using *Kodi* as a media centre.

Google Picasa

If the idea of hosting your photos online in order to make them accessible sounds appealing, then Google's Picasa service (now part of Google+) will appeal (assuming you like to use Google products, of course). You get up to 15GB free online storage – shared with Google Drive and Gmail – and it bends over backwards to make your photos as accessible as possible across a variety of platforms.

Despite the fact Google officially dropped support for the *Picasa* photo-editing tool on Linux some time ago, you can technically install it via *Wine*, but crucially the upload feature doesn't work, so while you might be able to prep your photos using its tagging tools, you can't then share them that way. Thankfully, you can connect to your Google account via other applications: a good example is *Shotwell* (<http://yorba.org>).

Once your library is up and running, simply select the files you wish to share, then choose File > Publish. From here select your Google account, then click Login to link your account to Shotwell Connect. From here you can choose which album to upload your photos to, as well as set a preset size for those photos and choose whether or not to include identifying information before uploading. Finally, click Publish and your photos will be uploaded and automatically published.

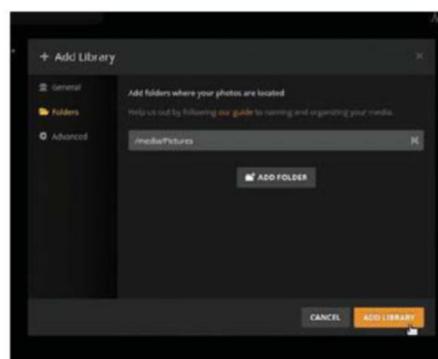


Use Shotwell to upload photos from your server to your Google account.

Going forward, you can access your Picasa Web Albums from other devices – you could, for example, open a desktop web browser and visit <https://plus.google.com/photos>. If you have an Android phone or tablet, you'll already know that your camera snaps can be automatically backed up, and that you can manage them using the built-in *Gallery* app. If you're looking for something a bit more sophisticated, try Colifer Lab's clumsily titled *Tool for Picasa, Google+ Photo* app (<http://bit.ly/ToolforPicasa>) which also supports

casting your photos to Chromecast. If you have an iPad or iPhone, we recommend investing £2.49 in the *Web Albums* app from Pixite (<http://webalbumsapp.com>), which lets you automatically back up photos from your iOS device as well as browse and manage your Picasa Web Album photos directly from the app. It also supports streaming photos to Apple TV via AirPlay. You should then scour your other connected devices to see whether a dedicated Picasa app is available – both the Roku and Now TV boxes offer one, for example. ➤

Add a photo library in Plex Media Server



1 Install Plex

Don't install *Plex* via your software repository – it's likely to be out of date. Instead, download and install the latest version from <https://plex.tv/downloads>. Once done, launch the app. If it doesn't appear in your web browser, type: **localhost:32400/web** to access it, then agree to the licence terms and click the + button next to your computer name to add your first channel.

2 Add your photos

Select Photos from the list of available options and click Next. Click Add Folder to add the first folder containing your photos – click Add when selected. You can add multiple folders around a specific theme (such as Holidays or Family History) or add your main Pictures folder and click Add Library to continue. Wait while the photos are added to your library.

3 Edit your library

By default photos are organised using their original file or folder names. Edit folder names by simply rolling the mouse over a thumbnail and clicking the pencil icon that appears. You can rename the title, create a Sort Title to organise it in a different order, and provide a description. Edit individual photos by clicking one to select it, then roll the mouse under it to reveal the editing tools.



Access media on devices

Let's connect some devices to your shiny new media server.

Now your *Kodi* media server is up and running, you're in a position to start enjoying your content anywhere in your home. Now all you need to do is find out how to get all the various devices – your other computers and laptops, smartphones, smart TVs and set-top boxes – talking to *Kodi*. We've got all the major bases covered, so read on to find out how to watch *Arrested Development* episodes on any device.

» **Other desktops and laptops** The obvious way to access your media server's library on another desktop or laptop is to simply install *Kodi* on that computer. Once done, open it up and select Videos > Files > Add Videos and Click Browse and select UPnP Devices. Select the *Kodi* server entry and click OK to add the entire library as a single source. You can then browse and access your media as if you were sat at the server itself – simply select the server from the list of available folders. *Kodi* can also access your *Plex Media Server* content too – we recommend installing the *PlexBMC* add-on for the best experience.

» **Raspberry Pi** The popular tiny board is a versatile device at the best of times, so we'd suggest using one as a dedicated media centre underneath one of your TVs. Accessing your *Kodi* media collection is simple thanks to

Raspbmc, which you can install to a SD card in Linux following the very helpful guides at www.raspbmc.org. Once done, pop the SD card into your Pi and boot from it to access and enjoy your media.

» **Android phones and tablets** You'll find a wide range of UPnP-compatible media players for Android, but if you want to access your media outside of your home network and need transcoding capabilities, you will need to have installed *BubbleUPnP* (see p36). Another option is to take your media with you – the *VidOn Player* app is free (<http://bit.ly/VidOnPlayer>), plays a wide range of media and even enables you to download content to your mobile for watching offline while you're away.

Another alternative, if your phone or tablet has enough grunt, is to install the official *XBMC for Android* app – this isn't the *XBMC Remote* you'll find on the App Store, but a full port of *XBMC* itself, accessible via your Android mobile and practically identical in looks and functionality to the desktop version.

To get *XBMC for Android*, first verify Android is set to enable third-party installations – tap Settings followed by Security and scroll down to tick Unknown sources under Device Administrators. Then open your browser and go to <http://kodi.tv/download> before downloading the correct version for your

device (which will be the ARM one in all but a handful of cases). It's a hefty download, so be patient, then simply tap the downloaded file and follow the prompts to install it.

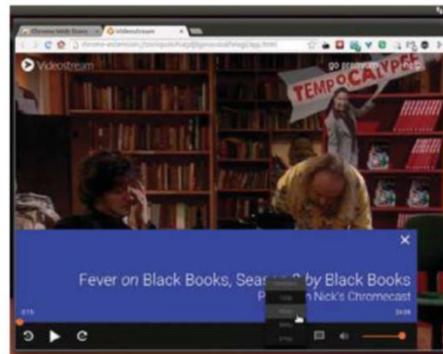
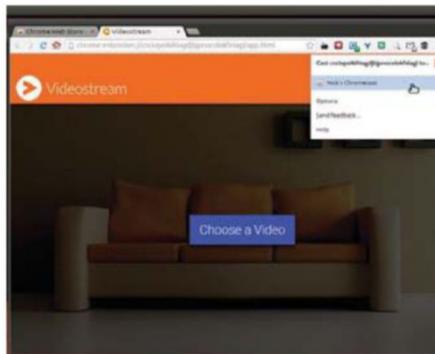
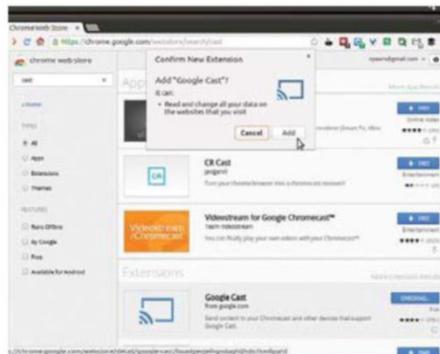
From here, simply follow the same instructions as for *Kodi* on the desktop to add your server as a media source. Note: it can be significantly slower for servers to appear than when using *Kodi* on a laptop or desktop, so be patient. Also be prepared for streaming issues when sending large files – such as high-definition movies – over your wireless network.

» **Apple iPhone and iPad** There's no *XBMC/Kodi* port for Apple devices, which means you're forced to look at a third-party media player for your needs. *Media Link Player Lite*



» **Kodi's the perfect media centre client to install on your other desktops and laptops.**

Cast video from your server to Google Chromecast



1 Get extensions

Launch Chrome and go to the Chrome Web Store (<http://chrome.google.com/webstore>). Search for the word **Cast** under Apps and install Google Cast, followed by *Videostream for Google Chromecast*. Once installed, *Videostream* will automatically open, but for future reference you can access it via Chrome's Apps menu (type `chrome://apps` into the Address bar).

2 Select your device

Click the Cast extension button in the browser toolbar and select your Chromecast device. Next, click Choose a Video – remember, *Videostream* doesn't work directly with *Kodi* or your UPnP server; instead, you need to manually select videos from your local or network storage. Browse to a file, open it and the file should start playing automatically through Chromecast.

3 Performance tweaks

Playback is controlled through your browser window – if the quality or performance isn't up to scratch, click the Quality button and choose a lower resolution to see if that improves matters. Tap the Settings button to the right of the Help button for other preferences, and visit <http://community.getvideostream.com> to get more assistance with issues should you encounter them.

(<http://bit.ly/MediaLinkPlayerLite>) is the free option, which provides a functional – if unspectacular – interface, but it does the job.

Also keep an eye out for *VLC for iOS* (<http://bit.ly/VLCForIOS>) – at time of writing it wasn't available on the App Store, but the developers were building a new iOS 8-friendly version that supports the later iPhone models, so expect it to reappear shortly. Like its desktop counterpart, *VLC for iOS* can play just about any format, and you can access your *Kodi* media server via the Local Network menu entry.

Another app worth trying is *8player lite* (<http://bit.ly/8Player>), which supports a wide range of video and audio codecs and provides a reasonably intuitive interface through which to browse and access content from your *Kodi* server. You can upgrade to the paid version for £2.99 and you gain the ability to access your *Kodi* server remotely.

» **Apple TV** If you have a jailbroken Apple TV or Apple TV 2 device, you can install *Kodi* on it – see the *Kodi* wiki (<http://kodi.wiki>) for details. Once done, it behaves like every other instance of *Kodi*. If you're looking to stream your shared media to an Apple TV that hasn't been jailbroken, then you'll need to find some way of using AirPlay to stream content to your Apple TV. That basically means pressing your mobile into service as an intermediary device, then finding a suitable app that can stream content from your media server while outputting it via AirPlay to your Apple TV.

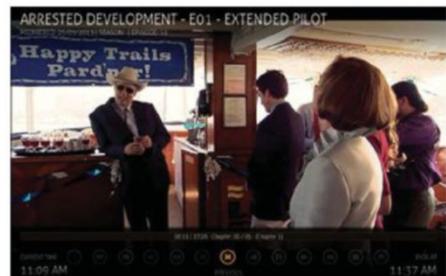
Obviously iOS users are well catered for here – AirPlay support is built into most apps, including *Plex*, *Media Browser* and *8player*.

And the good news for Android users is that *VidOn Player* provides AirPlay support on that platform too. When using *VidOn Player*, tap Settings followed by AirPlay and switch to Compatibility Mode to enjoy the best results when streaming media to your Apple TV.

» **Smart TVs and Blu-ray players** So long as your device in question has support for streaming media from DLNA/UPnP servers, you should be able to access your media server from the comfort of your armchair. The big issue will be with format support – if your media isn't encoded in a format the smart device recognises, you'll need to transcode it. These days, the standard format tends to be the x264/H.264 codec in a MP4 container with MP3 audio. If you've set up *BubbleUPnP* and have an Android device, you can use that to handle the transcoding.

» **Roku/Now TV** The Roku media player is again limited in its format support, so don't be surprised if the built-in MyMedia channel for accessing UPnP-shared content fails to work without the help of *BubbleUPnP*. Meanwhile the cut-down – but incredibly cheap – Now TV box comes with the PlayTo app that enables you to push content from your *Kodi* server via an Android device running the corresponding app, but it's fiddly and not very reliable.

If you're a *Plex* or *MediaBrowser* user, however, the Now TV box has one badly kept secret that's a game-changer: Developer Mode. Once enabled, this allows you to 'sideload' one additional app on to your Now TV box through your computer's web browser – the step-by-step guide below reveals how to add *RARflix* to



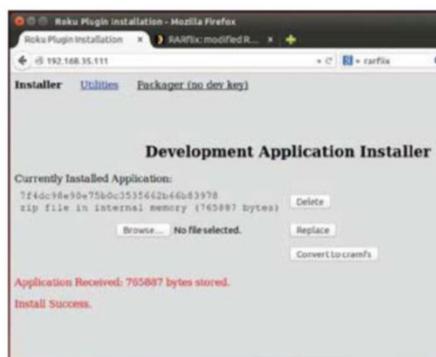
» **You can run XBMC on your Android device just like you can on your desktop PC.**

your Now TV box for accessing your *Plex* server, but you can also add *MediaBrowser 3* (<http://bit.ly/howtv-mb3>).

» **Google Chromecast** This only supports WebM and MP4/H.264 video, so it's likely you're going to need transcoding support. *Plex* is the simplest option for both Android and iOS, but *BubbleUPnP* can also transcode into the desired format if you're an Android user. Another option if you've access to *Chrome* (but not *Chromium*) through your media server is to install the *Videostream for Google Chromecast* plugin from the Chrome Web Store, which transcodes as well as beams media direct to your Chromecast device – it's less elegant than other solutions in that you need to select the file rather than browse your media library, but the step-by-step guide (see p40) reveals all you need to know. And if you like the idea of Chromecast, but don't like the fact it's part of Google, keep an eye on the crowdfunded Matchstick (www.matchstick.tv), which is similar concept to Chromecast but built on Mozilla's Firefox OS. LXF



Access a Plex media server through a Now TV box



1 Developer mode

From your Now TV box's remote, press the Home key three times, then up twice, then right, left, right, left and finally right again. You'll see a screen similar to that shown above – select Enable installer and restart, read the licence agreement and click I agree, then choose Enable installer and restart again. After rebooting, choose a password to prevent unauthorised access to your box.

2 Download and install

Open a browser window and enter your Now TV box's IP address. Enter **rokudev** as the username and your password to log on to the box. Now open a separate tab and browse to <http://bit.ly/rarflix-nowtv>, scroll down and click the link to download *RARflix*, saving the ZIP file to your Downloads folder. Switch back to the original tab, and click Browse to select the ZIP file, followed by Install.

3 Enjoy your media

Once installed, read the welcome message on your Now TV box and *RARflix* will then immediately go looking for your *Plex* media server, giving you access to all your media content – plus any additional web channels you've set up through your £10 Now TV box. And unlike the official *Plex* Roku app, *RARflix* is completely free (although donations are always welcome) to install and use.



A flight of fancy

Russell Barnes talks to the small team of Swedish entrepreneurs behind Crazyflie 2.0: a tiny open source quadcopter-cum-development platform designed to help your hardware hacks take flight.



Do you like the idea of an open source development platform that flies? Tobias Antonsson, Marcus Eliasson and Arnaud Taffanel do, and they've been perfecting the formula since 2009. After scoring a hit with their first palm-sized nano quadcopter, the trio – collectively known as Bitcraze – left their day jobs to develop the 92x92x29mm Crazyflie 2.0 full time. We caught up with them to see why we should give a flying FLOSS about their \$180 device.

Linux Format: Where did the idea of making a tiny open source quadcopter come from?

Tobias Antonsson: Really boring day jobs [laughs]. We were all working as consultants and we happened to be employed by the same customer at the time. The company we were working for had a system in place where you could pitch a project idea and they would help you by sponsoring it. They would buy hardware, that kind of thing. Unlike most set-ups where any ideas you have automatically become the property of

the employer, with this system you were able to keep the intellectual property. We started our first design towards the end of 2009 by doing the development in the evenings and we got our first prototype off the ground within six months.

Marcus Eliasson: This was right back at the start of the quadcopter era. At that time there were no small quadcopters. To start with we actually planned just to build a normal-sized model, because we wanted to try out the technology. But we ultimately decided to try and do something that nobody had done before, so we decided to make it as small as we possibly could and Crazyflie 1.0 was born.

Arnaud Taffanel: All our expertise is in the field of electronics, so we tried to make it as simple as possible. The original version had the motors glued onto the PCB itself – we've basically been trying to make a PCB fly.

LXF: Having the PCB act as the chassis for the quadcopter – motor arms and all – is a really clever design. What challenges were involved in getting it off the ground?

TA: In hindsight, I think one of the things that slowed us down with the original Crazyflie was that it was designed and built before 3D printers



were widely available. Since we were having to do things like glueing the motors to the PCB itself, every time it crashed we had to replace and re-glue the motors. Debugging in those conditions isn't easy. In the very beginning, Crazyflie didn't fly very well at all, so it took forever to find simple bugs because it would crash, then you would have to spend ages repairing it and then it would just crash again. It took a while, but we eventually settled on a technique that worked: we glued on a drinking straw so we could keep [the device] in the air long enough to spot the bugs.

LXF: If building one of the first nano quadcopters wasn't hard enough, you wanted to do it entirely with open source tools. Why was that so important to you?

TA: I think part of us just wanted to see if it was possible to make an advanced device like this using only open source tools. When we started in 2009 the applications that we used to create the circuit board diagrams, like *KiCad* (www.kicad-pcb.org) were nowhere near as well developed as they are now. There was definitely some struggling involved as we worked around bugs and so on.

ME: We've always been fans of open source [so] ultimately, the goal was to create an open source project. Since it's also an open source development platform, it's really important that our users have access to the same tools that we used to create it. If we used proprietary tools to design it, there's nothing to say that everyone in the community would have access to them. Using software like *KiCad* means that everyone that wants to can look at and play with our design files.

AT: It also allowed us to make a virtual machine of our entire development environment. Embedded system development can be really complicated at the best of times: you have to install all the tools, the compilers and the cross compilers. With everything being open source

we were able to make an Ubuntu virtual machine with all the development tools and the Crazyflie software pre-installed. It's quick and easy to download and use on just about any computer too: something that would be completely impossible with a proprietary toolset.

LXF: You're currently in the process of releasing the second iteration of your quadcopter. Is 2.0 all about fixing faults with

ON CRAZYFLIE 1.0

“The original version actually had the motors glued onto the PCB itself.”

the original design, or have you built it with specific new features in mind?

TA: Some parts of the new model are about correcting things from the original design, but we've mainly tried to create a more versatile flying platform that can handle a lot of different use cases. There's growing interest in developing drones and what they'll be able to do in the future, so all-round versatility has been really important to us.

ME: One of the first things we wanted to do was add an expansion port. All of the unused capability of the processor on the original Crazyflie was routed to an expansion port and we told people to attach whatever they wanted. Our users added lots of things like new devices and sensors, but it was really hard. You had to solder [the hardware] and make the drivers yourself – it wasn't an easy process.

With Crazyflie 2.0 we've made a much better expansion port – much more plug-and-play. We're planning to sell expansion boards too, the idea being that users can expand the capabilities of their platform really easily, just by clipping them in. Users can also make their own expansion boards and we're trying to lower the barrier to entry by making them easy to develop.

LXF: The Crazyflie 2.0 fits in the palm of your hand. How much weight can it carry before it falls out of the sky?

TA: One of the areas of improvement in Crazyflie 2.0 is that it can carry much more weight. We went from 5 grams from the original model to 15 grams for 2.0. It's not a huge amount, but more than enough to carry a small



» camera or a GPS device. We've got bigger motors and a bigger battery to thank for that, which means better flight performance too.

LXF: What other new features does Crazyflie 2.0 have?

ME: We've added a sleep mode meaning the new model can stay asleep for up to a year before the battery drains. While that's probably of little interest to people who just want to use the Crazyflie as a toy, you could spread lots of quadcopters around an area and have them wake up at different intervals to carry out various tasks. We've tried to pack as much functionality into the hardware as possible, so there's plenty of scope for innovation.

ME: We've also got [support for] Bluetooth low energy. With Bluetooth LE you can essentially control the Crazyflie 2.0 from your Android device or iPhone.

AT: Bluetooth LE is mostly about lowering the barrier to entry for people that just want to have fun with Crazflie. We already have prototypes of Android and iOS Crazyflie control applications working and we'll publish for both platforms on their corresponding markets.

ME: We also have a new Crazyradio. Crazyradio PA is a PC USB dongle that allows you to control the Crazyflie from your PC using an Xbox or PlayStation controller. This has a range of up to one kilometre with line of sight and good conditions. The old Crazyradio is still compatible too, though.

LXF: Aside from enabling you to control Crazyflie from a PC, what other benefits does the Crazyradio PA have?

ME: It's also a brilliant low-latency way of gaining low-level access to the Crazyflie itself.

The protocol for the USB dongle and the radio protocol are fully documented on our wiki.

Using the Crazyflie from your computer gives you access to adjustable flight software and a lot of development APIs too. We use Python, but there are APIs for C, C++, Java, JavaScript, Ruby and so on. It all works with Mac, Windows and Linux.

LXF: What sort of people are you expecting to buy the Crazyflie 2.0?

ME: One of the key issues we have is that the potential market is incredibly wide: from people using it as a toy to people using it as either a development platform or a research tool.

Straight out of the box there's loads of scope to have fun with it as a toy. You can unpack it and assemble it within a couple of minutes and then start flying it with your iPhone or Android device. Unlike the original Crazyflie, there's no soldering needed, which is going to be great news for a lot of more casual users. It's really fun to fly and the performance is insane.

Once the novelty of flying it around your house has worn off there are endless design and development possibilities, which you can learn step by step. It's essentially a flying development platform.

At the other end of the spectrum we have researchers using Crazyflie. They're not particularly interested in controlling it from their iPhone with Bluetooth LE ... they are more interested in what open source libraries we support.

As an open source development platform, Crazyflie has a really staggering reach. You can do iPhone development, Android development, embedded C, Python ... you can even create your own APIs for the PC version. For anyone

who wants to experiment with flying robotics, it's a brilliant platform that allows you to do whatever interests you.

LXF: What is it like to fly? The first one took a bit of getting used to.

TA: I know this is going to sound cocky, but it's really hard to explain how much fun it is to fly or how good it feels when you get the hang of flying it. It's incredibly fast and precise, and a side effect of [the increased] carrying weight is that it's much more aggressive in the air. It can lean 40 degrees in its direction of travel and before you know it it's almost out of sight.

ME: It takes four or five flights to really get the hang of it, but [that figure] depends who you're talking to. If you give the controller to anyone under the age of 20 and you start explaining what each button on the controller is for they've already hit the ceiling, but older users are much more careful. Despite all our practice we've still been put to shame by 15-year-olds that play a lot of PlayStation games [laughs].

AT: In the software on the PC you can tune all of the control parameters. When someone flies it for the first time they are not flying with the same aggressive settings as we would, for instance. We tune down the thrust, how fast it rotates and how far you can angle it.

One of the things about the original Crazyflie was that you were really quite afraid to crash it. The Crazyflie 2.0 is a lot more robust and you're much less afraid. The new motor mounts give it lots of protection – it actually bounces now.

LXF: How does the expansion port work? Are users going to be able to buy add-on boards separately?

TA: We spent a lot of time on the expansion port at the beginning of the development cycle. We wanted to be able to mount expansion boards underneath and on top of the Crazyflie 2.0. We needed some kind of system that effectively allowed us to mount an infinite number of boards, though, of course, weight and balance come into play quite quickly.

We looked at a lot of different systems and connectors and we came up with a solution that raises the expansion boards above the battery. It's one of the key features of the new model – the fact you can really quickly and easily adapt the Crazyflie to your needs.

Of the add-ons we have planned, right now we have a wireless charger that uses the same technology as a lot of modern smartphones and an LED ring with lots of RGB LED lights that connects to the bottom of the Crazyflie and faces downwards. It creates amazing effects in low light.

LXF: What if you want to develop your own expansion boards?

TA: We also have a breakout expansion that will allow you to mount your Crazyflie on a



prototyping breadboard to plan out add-on ideas. And we have a prototyping expansion board that you can solder your own hardware to and get your idea straight into the air.

ME: For example, we made a flying traffic light to demo the prototyping board. We soldered three coloured LEDs on in traffic light formation, one above the other, and we wrote some software to control them.

It's a very basic example, but you could do just about anything you like. You could easily make a small flying speaker by soldering a small piezo speaker to a prototyping board and fly around playing annoying noises at people. We're also working on different ready-to-use expansion board ideas. We have a board for GPS and a prototype for a camera. We're also working on boards with an SD card for logging large amounts of data.

LXF: Are you doing anything to facilitate community development of the boards?

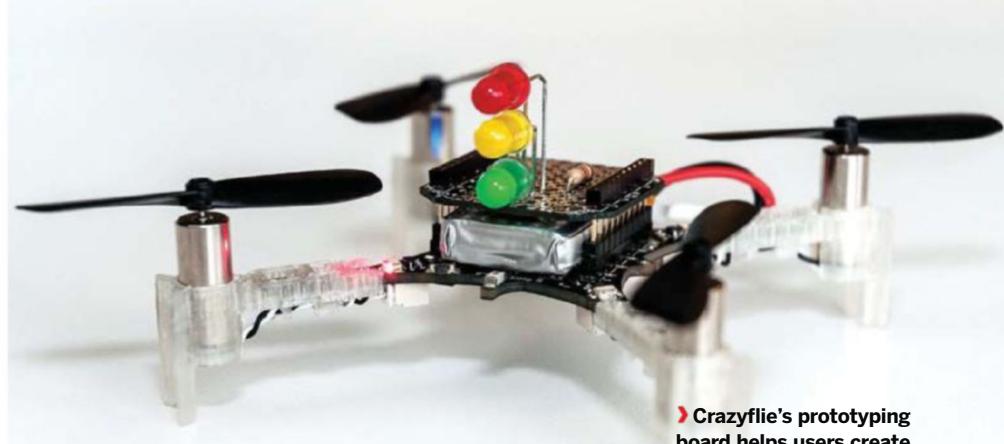
TA: We're really hopeful the community will come up with lots of ideas. We've already made a template for creating your own expansion boards and that's going to be added to the development platform virtual machine.

ME: If you open up the project in KiCad it already has the schematic components for the expansion connector so you can use KiCad's library to pick and place whatever hardware you want there. We've placed the expansion connectors so you only have to worry about wiring your own hardware.

We've also drawn in the Crazyflie 2.0 around it, so you can immediately see what kind of clearance you need for the connectors and for the propellers. You'll get a really good idea of what [the board] would ultimately look like when it's finished.

LXF: Are there any plans to make Crazyflie 2.0 capable of autonomous flight?

AT: One of the biggest challenges with trying to fly something like the Crazyflie autonomously is being fully aware of where you are in 3D space. By using things like Microsoft Kinect we developed a very simple solution and from that the community made some even better versions. Using Kinect you can certainly make



► **Crazyflie's prototyping board helps users create their own add-ons, like this flying traffic light.**

Crazyflie runs completely autonomously, but obviously only within sensor range of the hardware.

We're also collaborating with a company that builds simple optical flow sensors. Hopefully in the future we'll see an add-on board that might make it possible to direct a Crazyflie with scripted commands like 'go up one metre' or 'forward one metre'.

LXF: What about GPS as a means to get reasonably accurate location data, as with bigger quadcopters?

TA: It's obviously much simpler with GPS, but Crazyflie is focused on indoor applications. It can fly outside quite happily, but it isn't designed to cope with harsh weather conditions. The fact that you use it inside is one of the key reasons why Crazyflie is proving so popular with researchers. When you're working

modifying everything in the quadcopter. It should be as easy as adding one or two files – that's what we're working on.

We've been inspired by the way Linux and Android drivers work. We're essentially trying to solve the same kind of problem. In Linux when you have a device driver, you're not modifying files all over the place; you're modifying one file that automatically gets registered and can request access to different resources. We want a system where your file would declare that it is a driver for the LED ring expansion board, for example, and that it needs access to PWM [Pulse Width Modulation].

LXF: Manufacturing is often the most difficult part of a hardware-based project.

How are things working out for you with Crazyflie 2.0?

TA: We're partnered with a Chinese company called Seeed Technology (www.seeedstudio.com)

ON THE UNIVERSITY OF BOLOGNA

"They're using our platform to research search and rescue algorithms for alpine rescue."

with sensitive lab equipment you really need to be in a controlled environment.

For example, a research lab at the University of Bologna has worked on getting the Crazyflie to work autonomously with their very expensive tracking system. They're using our platform to research search and rescue algorithms for alpine rescue.

that works on a lot of hardware projects like this one. At the moment we're spending a lot of time making sure they're buying the right components and making sure everything is the right quality.

After that, just making sure that every Crazyflie 2.0 quadcopter that leaves the factory works is a huge responsibility. We actually all headed out to China so we could test the first units for ourselves.

Travelling out there to see everything first-hand has made things a lot easier ... for one thing, communication is much more straightforward in person. Working with Seeed has been great, though, because they're a company that's already used to working with small western companies like us, so the process is a little easier than it could be. **LXF**

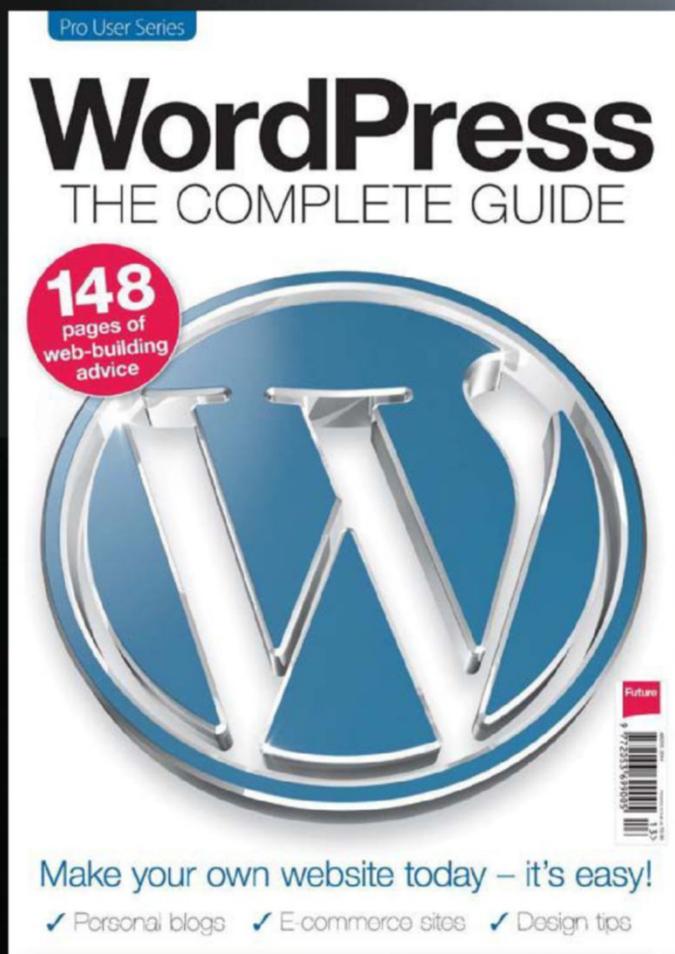
Crazyflie 2.0 is currently available to pre-order and should ship shortly after this issue of the magazine goes on sale. For up-to-date information, visit Bitcraze's website: www.bitcraze.se



Ultimately, we want someone to be able to have an idea in the morning, prototype it, and have written all the code and everything they need to make their idea actually fly by the end of the afternoon. That's the dream, but to make it that easy for the user requires a lot of work at our end.

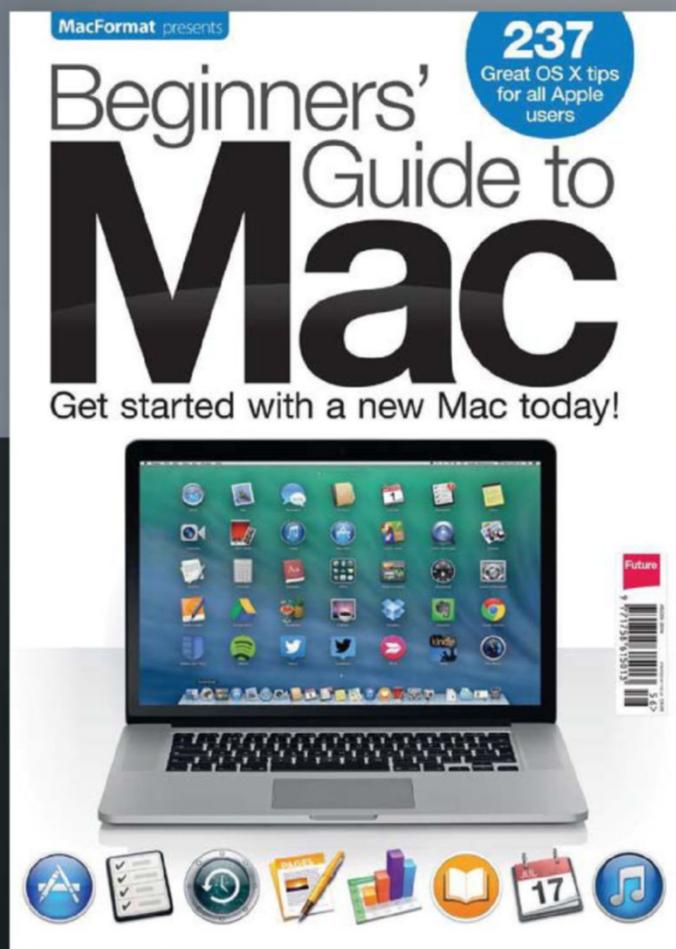
The way we see it, adding code and drivers for a new expansion board should not involve

CHECK OUT MORE GREAT BOOKAZINES!



WORDPRESS: THE COMPLETE GUIDE

Get started in web building with the complete guide to learning WordPress – the world's most popular website platform. Whether you're building a personal blog, portfolio site or commercial web store, you'll find WordPress to be the perfect tool for many years to come. We show you how to get started the easy way.



BEGINNERS' GUIDE TO MAC

If you're new to the Mac then this 164-page guide is the perfect starting point. There's dozens of great tips and tricks for the Mavericks desktop and we'll introduce you to the core apps that every new Mac comes with. When you're ready to go further we'll show you how to tweak your machine and become a power user fast!

ON SALE NOW!

DELIVERED DIRECT TO YOUR DOOR
ORDER ONLINE AT WWW.MYFAVOURITEMAGAZINES.CO.UK

FIND US IN YOUR NEAREST SUPERMARKET,
NEWSAGENT OR BOOKSTORE!

THE TECHPRO SERIES

Windows 8.1

Beyond the manual



377 Power tips – get more from your PC today!

Discover new ways to work and play with Windows 8.1

Perfect for PCs, tablets and laptops!

Future 9 772052 009997

THE TECHACTIVE SERIES

Raspberry Pi

MADE EASY



Get the most from the ultimate mini PC

196 plain English ideas & tips

FULLY REVISED & UPDATED FOR 2014

GET STARTED CORE SKILLS HOW TO CODE FUN PROJECTS

Future 9 772052 009997

WINDOWS 8.1 BEYOND THE MANUAL

Windows 8.1 Secrets and Hacks is the perfect companion manual for Windows 8.1 owners. You've probably already got this free update and have mastered the 'big stuff' by now, but maybe you also have a few questions, or there are some things you're not quite able to do.

That's where we come in – let us help!

RASPBERRY PI MADE EASY

The Raspberry Pi is an amazing device. Small, cheap and versatile, it's a fantastic bit of kit for learning how to code. Not much bigger than a credit card, it can be used for a range of tasks as diverse as emulating classic games and hosting your own online photo album. Find out how!

Filesystems: the next generation

Jonni Bidwell provides a primer on ZFS and btrfs: two of the most talked-up filesystems around.



Last issue, we created a glorious NAS box with 24TB of drives set up as a RAID 6 array formatted as ext4 [see *Homebrew your own NAS*, p46, LXF192]. This issue, we'll show you how to set up an alternative filesystem.

While ext4 is fine for volumes up to 100TB, even principal developer Ted Ts'o admitted that the filesystem is just a stop-gap to address the shortcomings of ext3 while maintaining backwards-compatibility. Ext4 first appeared in the kernel in 2008; up until then the most exciting filesystem around was ReiserFS. It had some truly next-gen features, including combined B+ tree structures for file metadata and directory lists (similar to btrfs). However, interest in this filesystem flagged just a touch when its creator, Hans Reiser, was found guilty of murdering his wife. Development of its successor, Reiser4, continues in his absence, but the developers have no immediate plans for kernel inclusion.

However, we now have a new generation of filesystems, providing superior data integrity and extreme scalability. They break a few of the old rules too: traditional ideologies dictate that the RAID layer (be it in the form of a hardware controller or a software manager such as mdadm) should be independent of the filesystem and that the two should be blissfully ignorant of each other. But by integrating them

originally released in 2005 as part of OpenSolaris, but since 2010 this has been disbanded and Oracle's development of ZFS in Solaris is closed source. Open source development continues as a fork, but since ZFS is licensed under the CDDL, and hence incompatible with the GPL, it's not possible to incorporate support into the Linux kernel directly. However, support via a third-party

module is still kosher and this is exactly what the ZFS on Linux project (<http://zfsonlinux.org>) does.

This project is largely funded by the Lawrence Livermore National Laboratory, which has sizeable storage

requirements, so ZFS can support file sizes up to 16 exabytes (2^{24} TB) and volumes up to 256 zettabytes (2^{38} TB).

Being an out-of-tree module, ZFS will be sensitive to kernel upgrades. DKMS-type packages will take care of this on Debian-based Linux distros, Fedora, CentOS, and so on, but for other distros you'll need to rebuild the module every time you update your kernel.

“Interest in ReiserFS flagged when its creator was found guilty of murdering his wife.”

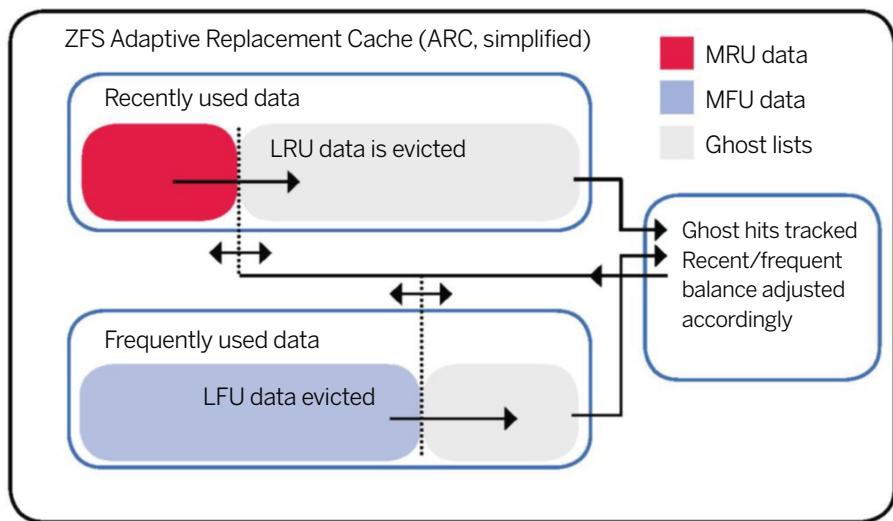
we can improve error detection and correction – if only at the cost of traditionalists decrying ‘blatant layering violations’.

The (comparatively) new kids on the block are btrfs (B-tree filesystem: pronounced ‘butter-FS’ or ‘better-FS’), jointly developed by Oracle, Red Hat, Intel, SUSE and many others, and ZFS, developed at Sun Microsystems prior to its acquisition by Oracle. ZFS code was

Failure to do so will be problematic if your root filesystem is on ZFS. Ubuntu users will want to add the PPA **zfs-native/stable** and then install the package **ubuntu-zfs**. The ZFS on Linux homepage has packages and information for everyone else.

Let's cover the common ground first. One quite startling feature is that neither of these filesystems require disks to be partitioned. In ZFS parlance you can set up datasets within a single-drive zpool which offers more isolation than directories and can have quotas and other controls imposed. Likewise you can mimic traditional partitions using subvolumes within btrfs. In both cases the result is much more flexible – the 'neopartitions' are much easier to resize or combine since they are purely logical constructs. ZFS actively discourages its use directly on partitions, whereas btrfs largely doesn't care.

Both of the filesystems incorporate a logical volume manager, which allows the filesystem to span multiple drives and contain variously named substructures. Both also have their own RAID implementations, although, confusingly, their RAID levels don't really tie in with the traditional ones: ZFS has three levels of parity RAID, termed RAID-Z1, -Z2 and -Z3. These are, functionally, the same



Caching in ZFS: two lists, for recently and frequently used data, share the same amount of memory. Most recently used (MRU) data is stored to the left and falls into the ghost list if not accessed. Memory is apportioned according to how often ghost entries are accessed.

would mirror the data twice, making for a usable capacity of 1TB. With btrfs, though, RAID 1 means that each block is mirrored once on a different drive, making (in the previous example) for a usable capacity of 1.5TB at the cost of slightly less redundancy. You can also use multiple drives of different sizes with btrfs RAID 1, but there may be some unusable space (hence less than half of the total storage present is available) depending on the combinatorics.

Additionally btrfs enables you to specify different RAID levels for data and metadata; ZFS features mirroring in much the same manner as RAID 1, but it does not call it that.

Mirroring with both of the filesystems is actually more advanced than traditional RAID, since errors are detected and healed automatically. If a block becomes corrupted (but still readable) on one drive of a conventional RAID 1 mirror and left intact on another, then mdadm has no way of knowing

which drive contains the good data; half of the time the good block will be read, and half of the time you'll get bad data. Such errors are called silent data errors and are a scourge – after all, it's much easier to tell when a drive stops responding, which is what RAID mitigates against. ZFS stores SHA-256 hashes of each block and btrfs uses CRC32C checksums of both metadata and data. Both detect and silently repair discrepancies when a dodgy block is read. One can, and should, periodically perform a scrub of one's next-generation volumes. This is an online check (no need to unmount your pools), which runs in the background and does all the detecting and repairing for you.

All this CoW-ing (Copy-on-Writing) around can lead to extreme fragmentation, which would manifest itself through heavy disk thrashing and CPU spikes, but there are safeguards in place to minimise this. ZFS uses a slab allocator with a large 128k block size, while btrfs uses B-trees. In both approaches the idea is the same: to pre-allocate sensible regions of the disk to use for new data. Unlike btrfs, ZFS has no defragmentation capabilities,



“Startlingly, neither of these filesystems require disks to be partitioned.”

as RAID 5, RAID 6 and what would be RAID 7, meaning they use 1, 2 and 3 drives for parity and hence can tolerate that many drives failing. RAID 5 and 6 are supported in btrfs, but it would be imprudent to use them in a production environment, since that part of the codebase is significantly less mature than the rest. RAID 0, 1 and 10 support is stable in both filesystems, but again the levels have a slightly different interpretation. For example, a conventional RAID 1 array on three 1TB drives

In the beginning, data was stored on punch cards or magnetic tape. The concept of a file didn't exist: data was stored as a single stream. You could point to various addresses in that stream (or fast-forward, using the tape counter to find where you recorded something), but it was all essentially a single amorphous blob.

Single-directory, or flat, filesystems emerged in the mid '80s. These enabled discrete files, but not subdirectories, to exist on a device. Their release coincided with increasing usage of floppy disks, which enabled random access of

data (you can read/write at any region of the disk). Early Mac file managers abstracted a hierarchical directory structure on top of a flat filesystem, but this still required files to be uniquely named.

By the late '80s filesystems that enabled proper directories were necessary to support growing storage technologies and increasingly complex operating systems. These had in fact been around since the days of IBM PC-DOS 2, but the poster child for this generation is FAT16B, which allowed 8.3 filenames and

volumes of up to 2GB. Windows 95 finally brought long filenames and the ability to access drives bigger than 8GB, but since 1993 Linux users had already seen these benefits thanks to ext2. This marked another step forward, featuring metadata such as file permissions, so that the filesystem becomes intrinsically linked with the user control mechanism. Ext3 and later revisions of NTFS introduced the next innovation: journaling, which allows filesystems to be easily checked for consistency, and quickly repaired following OS or power failure.

A brief history of filesystems

ZFS and btrfs

» which can cause serious performance issues if your zpools become full of the wrong kind of files, but this is not likely to be an issue for home storage, especially if you keep your total storage at less than about 60% capacity. If you know you have a file that is not CoW-friendly, such as a large file that will be subject to lots of small, random writes (let's say it's called **ruminophobe**), then you can set the extended attribute **C** on it, which will revert the traditional overwriting behaviour:

```
$ chattr +C ruminophobe
```

This flag is valid for both btrfs and ZFS, and in fact any CoW-supporting filesystem. You can apply it to directories as well, but this will affect only files added to that directory after the fact. Similarly, one can use the **c** attribute to turn on compression. This can also be specified at the volume level, using the **compress** mount option. Both offer zlib compression, which you shouldn't enable unless you're prepared to take a substantial performance hit. Btrfs offers LZO, which even if you're storing lots of already-compressed

data won't do you much harm. ZFS offers the LZJB and LZ4 algorithms, as well as the naive ZLE (Zero Length Encoding scheme) and the ability to specify zlib compression levels.

Note that while both btrfs and ZFS are next-generation filesystems, and their respective feature sets do intersect significantly, they are different creatures and as such have their own advantages and disadvantages, quirks and oddities.

Let's talk about ZFS, baby

The fundamental ZFS storage unit is called a vdev. This may be a disk, a partition (not recommended), a file or even a collection of vdevs, for example a mirror or RAID-Z set up with multiple disks. By combining one or more vdevs, we form a storage pool or zpool. Devices can be added on-demand to a zpool, making more space available instantly to any and all filesystems (more correctly 'datasets') backed by that pool. The image below shows an example of the ZFS equivalent of a RAID 10 array, where data is mirrored between two

drives and then striped across an additional pair of mirrored drives. Each mirrored pair is also a vdev, and together they form our pool.

Let's assume you've got the ZFS module installed and enabled, and you want to set up a zpool striped over several drives. You must ensure there is no RAID information present on the drives, otherwise ZFS will get confused. The recommended course of action is then to find out the ids of those disks. Using the **/dev/sdX** names will work, but these are not necessarily persistent, so instead do:

```
# ls -l /dev/disk/by-id
```

and then use the relevant ids in the following command, which creates a pool called **tank**:

```
# zpool create -m <mountpoint> tank <ids>
```

If your drives are new (post-2010), then they probably have 4kB sectors, as opposed to the old style 512 bytes. ZFS can cope with either, but some newer drives emulate the old-style behaviour so people can still use them in Windows 95, which confuses ZFS. To force the pool to be optimally arranged on newer drives, add **-o ashift=12** to the above command. You also don't have to specify a mountpoint: in our case, omitting it would just default to **/tank**. Mirrors are set up using the keyword **mirror**, so the RAID 10-style pool in the diagram (where we didn't have room to use disk ids but you really should) could be set up with:

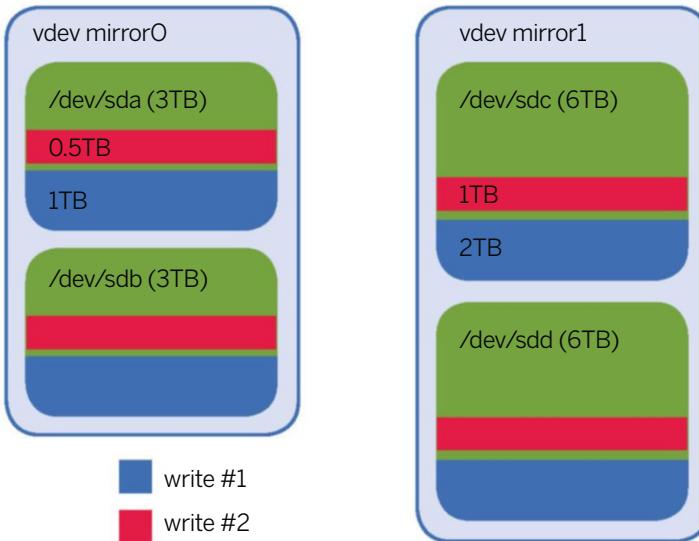
```
# zpool create -o ashift=12 mirrortank mirror /dev/sda /dev/sdb mirror /dev/sdc /dev/sdd
```

We can use the keyword **raidz1** to set RAID-Z1 up instead, replacing **1** with **2** or **3** if you want double or triple parity. Once created, you can check the status of your pool with:

```
# zpool status -v tank
```

You can now add files and folders to your zpool, as you would any other mounted filesystem. But you can also add filesystems (a different, ZFS-specific kind), zvols, snapshots and clones. These four species are collectively referred to as datasets, and ZFS can do a lot with datasets. A filesystem inside a ZFS pool behaves something like a disk partition, but is easier to create and resize (resize in the sense that you limit its maximum size with a quota). You can also set compression on a per-filesystem basis.

zpool



» **ZFS will stripe data intelligently depending on available space: after a 3TB write and then a 1.5TB write, all drives are half-full (or half-empty, depending on your outlook).**

Have a CoW, man

Even if you have no redundancy in your next-gen filesystem, it will be significantly more robust than its forbears. This is thanks to a technique called Copy-on-Write (CoW): a new version of a file, instead of overwriting the old one in-place, is written to a different location on the disk. When, and only when, that is done, the file's metadata is updated to point to the new location, freeing the previously occupied space. This means that if the system crashes or power fails during the write process, instead of a corrupted file, you at

least still have a good copy of the old one. Besides increased reliability, CoW allows for a filesystem (or more precisely a subvolume) to be easily snapshotted. Snapshots are a feature, or even the feature, that characterises our next-generation filesystems. A snapshot behaves like a byte-for-byte copy of a subvolume at a given time (for now think of a subvolume as a glorified directory – the proper definition is different for btrfs and ZFS), but when it is initially taken, it takes up virtually no space. In the beginning, the

snapshot just refers to the original subvolume. As data on the original subvolume changes, we need to preserve it in our snapshot, but thanks to CoW, the original data is still lying around; the snapshot is just referred to the old data, so the filesystem will not mark those blocks as unused, and old and new can live side by side. This makes it feasible to keep daily snapshots of your whole filesystem, assuming most of its contents don't change too drastically. It is even possible to replicate snapshots to remote pools via SSH.

Let's create a simple filesystem called **stuff**. Note that our pool **tank** does not get a leading **/** when we're referring to it with the ZFS tools. We don't want it to be too big, so we'll put a quota of 10GB on there too, and finally check that everything went OK:

```
# zfs create tank/stuff
# zfs set quota=10G tank/stuff
# zfs list
```

A zvol is a strange construction: it's a virtual block device. A zvol is referred to by a **/dev** node, and like any other block device you can format it with a filesystem. Whatever you do with your zvol, it will be backed by whatever facilities your zpool has, so it can be mirrored, compressed and easily snapshotted. We've already covered the basics of snapshots (see *Have a CoW, man*), but there are some ZFS-specific quirks. For one, you can't snapshot folders, only filesystems. So let's do a snapshot of our **stuff** filesystem, and marvel at how little space it uses:

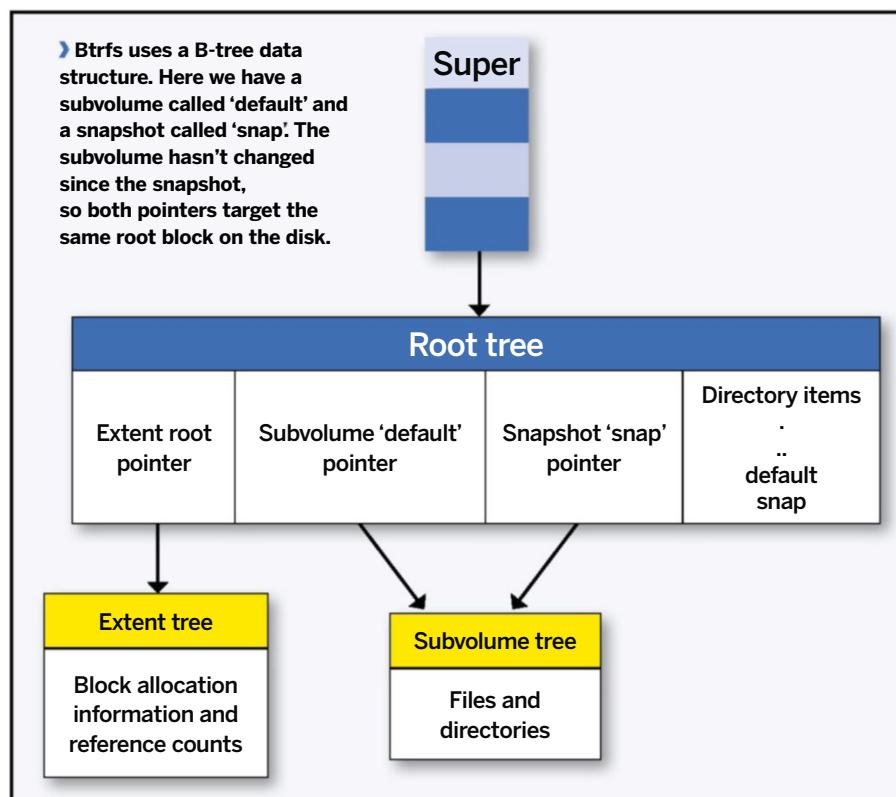
```
# zfs snapshot tank/stuff@snapshot0
# zfs list -t all
```

The arbase syntax is kind of similar to how a lot of systemd targets work, but let's not digress. You can call your snapshot something more imaginative than **snapshot0** – it's probably a good idea to include a date, or some indication of what was going on when the snapshot was taken. Suppose we now do something thoughtless resulting in our **stuff** dataset becoming hosed. No problem: we can roll back to the time of **snapshot0** and try and not make the same mistake again. The **zfs diff** command will even show files that are new (+), modified (M) or deleted (-) since the snapshot was taken:

```
# zfs diff tank/stuff@snapshot0
M      /pool/stuff
+      /pool/stuff/newfile
-      /pool/stuff/oldfile
# zfs rollback tank/stuff@snapshot0
```

Snapshots are read-only, but we can also create writable equivalents: the final member of the dataset quartet, called clones.

It would be remiss of us to not mention that ZFS works best with lots of memory. Some recommendations put this as high as a GB per TB of storage, but depending on your purposes you can get away with less. One reason for this is ZFS's Adaptive Replacement Cache. This is an improvement on the patented IBM ARC mechanism, and owing to its consideration of both recent and frequent accesses (shown in the diagram on p49) provides a high cache hit rate. By default it uses up to 60% of available memory, but you can tune this with the module option **zfs_arc_max**, which specifies the cache limit in bytes. If you use the deduplication feature then you really will need lots of memory – more like 5GB to the TB – so we don't recommend it. A final caveat: use ECC



memory. All the benefits offered by ZFS checksums will be at best useless and at worst harmful if a stray bit is flipped while they're being calculated. Memory errors are rare but they do happen, whether it's dodgy hardware or stray cosmic rays to blame.

Btrfs me up, baby

As well as creating a new btrfs filesystem with `mkfs.btrfs`, one can also convert an existing ext3/4 filesystem. Obviously, this cannot be mounted at the time of conversion, so if you want to convert your root filesystem then you'll need to boot from a Live CD or a different Linux. Then use the **btrfs-convert** command. This will change the partition's UUID, so update your fstab accordingly. Your newly converted partition contains an image of the old filesystem, in case something went wrong. This image is stored in a btrfs subvolume, which is much the same as the ZFS filesystem dataset.

As in ZFS, you can snapshot only subvolumes, not individual folders. Unlike ZFS, however, the snapshot is not recursive, so if a subvolume itself contains another subvolume, then the latter will become an empty directory in the snapshot. Since a snapshot is itself a subvolume, snapshots of snapshots are also possible. It's a reasonable idea to have your root filesystem inside a btrfs subvolume, particularly if you're going to be snapshutting it, but this is beyond the scope of this article.

Subvolumes are created with:

```
# btrfs subvolume create <subvolume-name>
```

They will appear in the root of your btrfs filesystem, but you can mount them individually using the **subvol=<subvolume-name>** parameter in your fstab or mount command. You can snapshot them with:

```
# btrfs subvolume snapshot <subvolume-name> <snapshot-name>
```

You can force the snapshot to be read-only using the **-r** option. To roll back a snapshot:

```
# btrfs subvolume snapshot <snapshot-name> <subvolume-name>
```

If everything is OK then you can delete the original subvolume.

Btrfs filesystems can be optimised for SSDs by mounting with the keywords **discard** and **ssd**. Even if set up on a single drive, btrfs will still default to mirroring your metadata – even though it's less prudent than having it on another drive, it still might come in handy. With more than one drive, btrfs will default to mirroring metadata in RAID 1.

One can do an online defrag of all file data in a btrfs filesystem, thus:

```
# btrfs filesystem defragment -r -v /
```

You can also use the **autodefrag** btrfs mount option. The other piece of btrfs housekeeping of interest is **btrfs balance**. This will rewrite data and metadata, spreading them evenly across multiple devices. It is particularly useful if you have a nearly full filesystem and **btrfs add** a new device to it.

Obviously, there's much more to both filesystems. The Arch Linux wiki has great guides to btrfs (<http://bit.ly/BtrfsGuide>) and ZFS (<http://bit.ly/ZFSGuide>). **LXF**

Mod Minetest

Jonni Bidwell explains how to tailor the open source Minecraft-inspired sandbox adventure to your tastes.

Since Microsoft's acquisition of Mojang in September, *Minecraft* players have become increasingly concerned and consternated about the future of both the game and the community (just check out this video: <http://bit.ly/MSMinecraft>). Notch and the gang have left, and are probably sipping mojitos in the Bahamas, wondering how to spend their \$2.5 billion. While Redmond has vowed to respect the passionate community that has *Minecraft* has engendered, nobody can say with any certainty what will be the future of the most popular video game in human history. Even if the new ownership doesn't bother you, it would still be nice if there was an open source alternative.

Oh, but there is. Minetest (<http://minetest.net>) is just that open source

(LGPL 2.1) Minecraft alternative. It's actually been around for about four years, and so has a relatively stable codebase, a dedicated community and its own subreddit ([/r/Minetest](#)).

Releasing open source versions of popular games is nothing new. A great example is *PrBoom* (<http://bit.ly/PRBoomGame>),

“Notch and the gang are probably sipping mojitos in the Bahamas.”

which is a rewrite of the *Doom* (and *Doom II*, *Heretic* and *Hexen*) engine. You can use the data files (which store maps, graphics and sounds) from your paid-for copy, or you can use community ones, such as those in the *Freedom* project (<https://freedom.github.io>). Other engine rewrites include *ScummVM*

(<http://scummvm.org>) for early LucasArts games, *OpenAge* (<http://bit.ly/OpenAgeGame>) for *Age of Empires II*, and *Stratagus Core* (<http://stratagus.com>) for *StarCraft* and *Warcraft II*.

Other projects have decided not to bother recycling or reverse-engineering original data, and opted instead for a complete rewrite. Being an open world sandbox type affair, it makes more sense to re-implement the *Minecraft* engine: there are no levels to speak of, and the sprites and textures are not the most onerous thing to redo.

The game has both survival and creative modes, and pretty much every aspect of it can be modded in the popular Lua scripting language. The *Minetest* core is mostly written in C++ and uses the Irrlicht (German for will-o'-the-wisp) engine. This makes the code highly portable, hence builds are available for Linux,

FreeBSD, Windows and OS X. It's also less resource-hungry than the Java-based *Minecraft*, so will run fine on older hardware (the developers recommend at least Intel 945 graphics and a dual-core processor). *Minetest* packages are available in the official repositories for Ubuntu, Debian (and Raspbian), Arch Linux, Gentoo, Fedora, OpenSUSE and probably many others. Download and compile the source code from the website in the unlikely event that your distribution is lacking packages.

Once you've gotten everything installed, you can run the game from the terminal with a simple:

```
$ minetest
```

You'll probably have an icon in your Games menu too. Start by creating a new world, think of an amusing name for it, click Create and then Play. Running and jumping will come to you instinctively, but you'll need to do more than that if you want to survive. Nightfall is coming, after all. Gathering wood is a reasonable first step, so find a tree and hack the trunk to bits with the left mouse button. As you embrace your inner lumberjack, you will see wood accumulating in one of the boxes at the bottom of the screen. This manner of harvesting applies to all the raw materials you might encounter in the game world, including, but not limited to, flowers, apples, gravel and sand. To refine your logs into planks press I to open the inventory window. Drag the logs into the nine-squared crafting area, and you will then be able to collect planks from the right-hand side. Planks can be further refined into sticks, which have a plethora of uses. There's much more to crafting than this though: you can make a pickaxe, for instance (pictured on p54), which will increase your mining prowess.

Bring the fire

There are many other things you'll need to craft in order to survive, chief among which is fire. Having the hot stuff, particularly in the form of torches, is very useful for brightening up the darkness. Torches require sticks, which we've already discussed, and coal, which you can gather using your recently manufactured pickaxe. More serious fire can be used for smelting in a furnace, which can be crafted by surrounding the centre square in the crafting grid with one of the many kinds of stone available. Just as in the original *Minecraft*, there are fixed recipes (such as the aforementioned pickaxe and furnace) which require ingredients to be arranged specifically in the crafting grid, and shapeless recipes, where the arrangement is not important. Once you've crafted a furnace, you'll want to find



This is the Liberty Land server, where users can claim land and leave letters in mailboxes.

some fuel to get it going, and something to smelt. Just like in real life, lots of things will burn, some better than others. Right-clicking on your furnace will reveal the smelting grid. Place some fuel in the lower square and something to refine in the upper square. After a few seconds, provided you have chosen suitable combustibles, the product will appear in the right-hand square. The complete guide to smelting can be found on the *Minetest* wiki (<http://bit.ly/MTSmelting>) though we were disappointed to find that rats, which used to provide a health-boosting in-game snack when cooked, are no longer part of the vanilla *Minetest* game.

In fact, when run as is, *Minetest* doesn't include any mobs (mobile entities) which might seem a little dull. But fear not, the Simple Mobs mod, which you find on the [LXFDVD](#), provides rats, sheep and five different kinds of scary monsters. To use it you'll need to make a directory called **mods** and extract the tarball there:

```
$ mkdir ~/.minetest/mods  
$ cd ~/.minetest/mods  
$ tar -xvzf simple_mobs.tar.gz
```

Now start the game and select Enable all from the Configure menu. The Mobs entry should turn green. When you open up a world, you might be lucky enough to find a sheep or two running around. You can even spawn one at your feet by pressing t (which will bring up the chat dialogue which also functions as a console) and typing:

```
/spawncity mobs:sheep
```

If you look inside the file **~/.minetest/mods/mobs/init.lua**, you'll see the other mobs provided: tree/sand/stone/dirt monsters, oerkki and dungeon masters, as well as the



The Minetest API

Minetest has three fundamental object classes:

► **Nodes** Blocks from the world.

► **Tools** Weapons, actual tools and other implements for making things.

► **Craftitems** Miscellaneous items (for example, sticks, paper and minerals).

When you come to write your own mods, you'll need to register any new objects you define in a file called **init.lua**, using, for example, the **minetest.register_node()** method. Any mod must contain at least this initialisation file, inside a directory contained either in

```
~/.minetest/mods for per-user mods, or  
/usr/share/minetest/mods for global ones.
```

As well as objects, you can also define crafting recipes. For example, if we add the following clause to the mobs mod's **init.lua** file, then we will be able to craft a Nyan Cat rainbow block using an apple and a block of dirt... just like in real life:

```
minetest.register_craft({  
    output = "default:nyancat_rainbow" 1',  
    recipe = {  
        {'default:dirt', 'default:apple', ''},
```

```
    },  
})
```

Nyan Cat blocks do occur naturally, but they're frightfully rare, and are only found 32 blocks beneath sea level. The **recipe** array corresponds to the arrangement of the items in the crafting grid. You can also specify an optional third parameter **type** which can be set to "**shapeless**", for a shapeless recipe, or "**cooking**" for furnace-based operations.



► The sun rises just before 5am, revealing the bones of a fallen player... which we plunder for loot.

less hostile sheep and rats. Perusing this file will also give you a taste of the Lua API which all mods use. For example, you'll see there's a method for right-clicking on a sheep which provides the player with wool, provided the sheep is not naked. You will also see that sheep will follow the player if they are carrying wheat. If you are feeling brutal, you can slaughter your sheep and cook it on the furnace for a health-boosting snack. You'll have to write your own mod if you want mint sauce, though.

With the Simple Mobs mod enabled, you'll notice that as night falls the bad things come out and set about ending your existence. Some of these are harder to kill than others, but a pickaxe will do more damage than your bare hands or, say, a leg of lamb. You might find yourself praying for

“There are many things you’ll need to craft in order to survive, chief among which is fire.”

daylight, though – and as luck would have it, your prayers can be answered, provided you can issue the command:

`/time 5000`

before your death at the hand of some ghoulish entity. The *Minetest* clock cycles from 0 to 23,999, with 4,500 more or less corresponding to sunrise, and the cessation of the spawning of monsters. Sadly, it won't stop already-spawned ones craving your flesh. If you want to cheat and give yourself a weapon without going through the arduous crafting processes, then issue the command:

`/giveme default:sword_diamond`

If the forces of darkness do overcome you, you will be able to respawn somewhere else. Should you return to the place of your demise, you will find your bones, through which you can rummage to reclaim all your stuff.

Pretty much everything you encounter in *Minetest* is done through mods – *Minetest* without mods is like a book

without words. When you activated the mobs mod earlier you probably noticed a few of them. The standard game mods are installed in the **games/minetest/mods** subdirectory of the install path. Here, for example, you can find out how flowers grow, fire burns and TNT detonates. The mod entitled default contains, amongst other things, the definitions for all the crafting recipes (in the file **crafting.lua**), all the weapon and tools (**tools.lua**) and all the available block types (**nodes.lua**).

In the menu for creating a new world, you might notice the Mapgen dropdown box. This provides a number of different algorithms for generating a map: v6 is the current default; v7 is experimental and will only produce stone landscapes unless you feed it some extra parameters. Procedurally generating landscapes is a complicated affair, requiring of a fine balance between randomness and consistency. The main technique involved is called Perlin noise which does a fine job of creating biomes: regions of similar climate, geography and ecology. If you have a very recent version (that is, if you compiled from source or are using Arch Linux), then you may also see some other experimental generators, such as math, which generates otherworldly 3D fractals.



► How to make pickaxes (and influence people). You can also use different materials for the blade.

There are a number of beautification options in the Settings menu which you should try out if your machine is powerful enough, including shaders, bump maps and waving water. Some might say these detract from the LEGO-like feel of the game, but we think they look pretty, and it's nice to challenge your video card. If you really want to jazz things up, you can download all manner of user-contributed texture packs from the *Minetest* forums (<http://forum.minetest.net>). Here you will also find many, many mods, messing with which is a great way to learn the ins and outs of the API.

Minetest-morphosis

One of the exciting features of the API is the ability to implement Active Block Modifiers (ABMs). These specify changes that can happen to nodes either over time or when they come into contact with nodes of a given type. For example, lava changing into cobblestone after coming into contact with water for one second is described in the file **functions.lua** by:

```
minetest.register_abm({
    nodenames = {"default:lava_flow"},  
    neighbors = {"group:water"},  
    interval = 1.0,  
    chance = 1,  
    action = function(pos, node, active_object_count,  
    active_object_count_wider)  
        minetest.set_node(pos, {name =  
        "default:cobblestone"})  
    end,  
)
```

The **chance** argument specifies the likelihood of the change taking place: **1** implies that it is certain, while higher values refer to the reciprocal of the probability, so **100** would be a one-in-a-hundred event. The **lava_source** node has a slightly more interesting reaction when it's cooled – besides turning to obsidian, it makes a satisfying noise. In order that both of these actions are executed, they are grouped together into a function, which looks like:

```
default.cool_lava_source = function(pos)  
    minetest.set_node(pos, {name = "default:obsidian"})  
    minetest.sound_play("default_cool_lava", {pos = pos,  
    gain = 0.25})  
end
```

Then the **action** parameter in the ABM registration is set to call this function:

```
action = function(pos, node, active_object_count,  
active_object_count_wider)
```



Our protagonist and his sheep enjoy some cooked rat by candlelight.

`default.cool_lava_source(pos, node, active_object_count,
active_object_count_wider)`

Just like the **getBlock()** and **setBlock()** functions, which form the crux of the oft-discussed *Minecraft* API, *Minetest* has analogous functions named **minetest.get_node()** and **minetest.set_node()**. Furthermore, there's also a very handy function, **minetest.find_node_near()**, which can find nodes of a given type within a specified radius of a given position. While you could code this yourself using the get methods, a couple of loops and a spot of Pythagoras, it's nice to know that you don't have to. This function is used, for example, to enable papyrus plants to grow when they are on top of grass and within three blocks of a water source:

```
minetest.register_abm(  
    nodenames = {"default:papyrus"},  
    neighbors = {"default:dirt", "default:dirt_with_grass"},  
    interval = 50,  
    chance = 20,  
    action = function(pos, node)  
        pos.y = pos.y-1  
        local name = minetest.get_node(pos).name  
        if name == "default:dirt" or name == "default:dirt_  
with_grass" then  
            if minetest.find_node_near(pos, 3,  
            {"group:water"}) == nil then  
                return  
            end
```

We'll go into much more depth about the Lua API in a future **LXF** tutorial, but hopefully this has served as an enjoyable introduction to the wonders of *Minetest*. May your blocky world provide all you need to survive and prosper. **LXF**



Multiplayer mayhem

Just like *Minecraft*, *Minetest* consists of both a client and a server. When you play in single-player mode your machine is set up as a local, private server and the client program connects to this. The server is in charge of all the game settings, including which mods are active. If you want to interact with other *Minetest* players, then pick a random server from the list at <http://minetest.net/servers>. These run in a variety of configurations: some will require a password, some run in creative mode, and some

will restrict privileges, such as the ability to chat, teleport or use the **/give** command.

You can run your own server, either on your local machine or a remote server. By default, these listen for UDP connections on port 30000, so you'll need to configure your firewall accordingly. You can start a dedicated server like so:

```
$ minetest --server
```

If you've got multiple worlds defined then you'll need to tell *Minetest* which one to serve,

either by path, with the **--world** argument, or by name, with **--worldname**. World data is stored in subdirectories of **~/.minetest/world**. You can set a variety of server-related options in the configuration file **~/.minetest/minetest.conf**. For example, if you want to advertise your server, set the options:

```
server_announce = 1  
serverlist_url = servers.minetest.net  
Consult the example file in /usr/share/minetest/doc/ for a complete rundown.
```



Dr Chris Brown

The Doctor provides Linux training, authoring and consultancy. He finds his PhD in particle physics to be of no help in this work at all.

Learning to type

One of the things I find astonishing about the computer professionals I meet is how few of them have actually learned to type. They spend most of their days interacting with computers through keyboards, but many of them do no more than peck with two fingers, laboriously scanning the keyboard for the characters they need. Their accuracy, too, often leaves a lot to be desired, and I sometimes worry at the thought of them sitting at a root command prompt on a production server.

I just measured my typing speed and I can manage about 50 words per minute, just typing stuff out of my head. (A 'word', in typing speed parlance, means five keystrokes.) An average professional typist usually manages 50-100. I'm significantly slower, at 30-40 wpm, when copying from printed text.

But there is a world of difference between being able to type English prose and being able to type fancy shell commands or program code, where there are all sorts of non-alphanumeric characters to contend with, upper-case characters in the middle of variable names, the need to take care over the indentation, and so on. I managed only 20 wpm entering a few lines of Perl code, for example. And importantly, the potential penalty of making mistakes is much higher. Consider the difference between typing:

`$ rm *.old`

and:

`$ rm *.old`

So I would be interested to know your views. How fast can you type? Do you think your poor typing skills, or poor accuracy, significantly limit your productivity as a system administrator? Or does the time spent thinking what you need to type dominate over the time spent actually typing it? Write to me: I'd love to hear what you have to say on the subject.

chris.linuxformat@gmail.com

Dr Brown's Administeria

Esoteric system administration goodness from the impenetrable bowels of the server room.



Cray usurps IBM at the Met Office

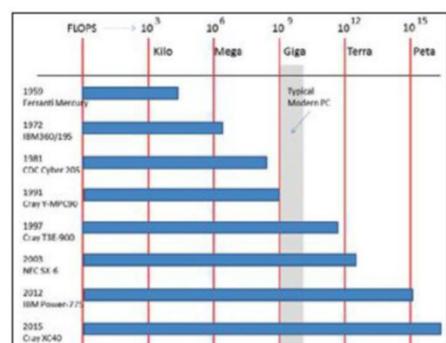
Bringing new meaning to the term 'cloud computing', the Met Office is shopping for more CPU cycles.

The Met Office, the UK's national weather service, is splurging £97 million on a new supercomputer – a Cray XC40 with go-faster stripes and titanium wheel trims. It almost goes without saying that it will run Linux. According to Cray's website, the "Cray Linux Environment ... includes a Linux-based operating system designed to run large complex applications and scale efficiently to more than 500,000 processor cores. The Linux environment features a kernel that can be configured to match different workloads." You can find the details at <http://bit.ly/CraySpecs>.

The Met Office is no stranger to supercomputing; it bought its first machine (a Ferranti Mercury) back in 1959. It could perform 30,000 calculations per second. By way of comparison, a modern PC is maybe 100,000 times as fast. But since then, it's acquired no less than seven supercomputers from IBM, CDC, Cray and NEC. The new machine, scheduled to become operational next September, will have 480,000 processor cores (Pentium Xeon) and claims a performance of 16 petaflops. I asked my wife what a petaflop was and she said she thought it meant that time in late summer when the flowers start to wilt, but in fact petal means 10^{15} , a number so improbably huge you risk a headache trying to imagine it. I have enough experience of exploiting parallel machines to

know that they're unlikely to get anywhere near that figure in terms of useful computation. Still, it's a big number, and 13 times as powerful as the IBM Power 755 the Met Office has now. According to my reading of the list of the world's most powerful computers (www.top500.org/lists), it's going to come in around number four, at time of writing.

How do you justify the cost of such a huge machine? Well, the Met Office claims it will deliver £2 billion of "socio-economic benefits" for the UK through "enhanced resilience to severe weather and related hazards" by providing more frequent forecasts and will enable strategically important areas, such as airports, to receive forecasts of wind speed, fog and snow to a spatial resolution of 300 metres.



► The Met Office has seen exponential growth in computing power over the last 50 years.

Set up an OpenLDAP server

The Doctor shows you how to centralise your user account information by setting up an OpenLDAP server on Ubuntu.

This morning's reading is taken from the book of Tux, chapter five, verse one. In the beginning was the password file, and the password file was with Unix. Through it, all users were logged in; without it, no one logged in that had logged out. And the sysadmins saw that it was good. But lo! there came the time of the Great Networking, and the sysadmins spake amongst themselves, saying, "The password file serveth not well, for it requireth replication of data and scaleth not to large networks." And the Sun said, "Fear not, for we bring you Yellow Pages, which centraliseth the user data."

But there came wise men from Beetea, saying, "Thou mayst not take the name Yellow Pages, for it has been registered unto us as a trade mark." So the Sun said, "Henceforth that which was known as Yellow Pages shall be called NIS." And the sysadmins saw that it was good.

But after a time, a disenchantment arose again within the sysadmins who complained a second time, saying, "Verily, NIS hath but a flat namespace, and no access control."

And again the Sun said, "Fear not, for we bring you NIS+, which hath a hierarchical namespace and access control in abundance." But the sysadmins complained a third time, because they comprehendeth it not.

And so it came to pass that a great consortium was created to draw up the X.500 specification. And X.500 begat DAP, and DAP begat DIXIE, and DIXIE begat LDAP. And the sysadmins saw that it was good.

Here endeth this morning's reading.

This month (rapidly dropping out of my vicar vernacular) we'll learn the basics of LDAP and see how to set up an LDAP directory service to store user accounts. Next month, we'll see – among other things – how to configure a machine to use an LDAP server as a source of account information.

An LDAP primer (just the first coat)

LDAP stands for Lightweight Directory Access Protocol, but generally when we talk about LDAP we also mean the server that actually speaks the protocol and stores the information in the directory. In principle, you could store any kind of information in LDAP, but in practice it tends to be used as a sort of enterprise-wide address book, holding user names, telephone numbers, postal addresses, email addresses, job titles and departments, and so on. In particular, it can store user account information – the sort of things that were traditionally stored in **/etc/passwd** and **/etc/shadow**.

An LDAP directory stores information in a tree structure, much like the file system does (or the DNS, for that matter). This tree is called a DIT (Directory Information Tree). Each entry in the tree is identified by a 'distinguished name': something like **uid=mary,ou=People,dc=example,dc=com**. The first part of this (**uid=mary**) is called the relative distinguished name and the rest is the distinguished name of the parent node (**ou=People,dc=example,dc=com**). This is roughly analogous to a full pathname within the Linux file system, such as **/home/chris/articles/ldap**, where **ldap** is the file name and **/home/chris/articles** is the path name of the parent directory. But notice that the components are in the opposite order – distinguished names are written

Backend storage

LDAP (as a protocol) defines a way to access data; it doesn't specify how it's to be stored. The default storage back-end is **hdb**, a variant of the venerable Berkeley DB indexed database. The actual files are in **/var/lib/ldap** by

default, but you can't examine these files directly in any meaningful way. You can also use the text-based LDIF format for back-end storage; this is what's done for the **cn=config** DIT, but you wouldn't want to use it for a large directory.

little-endian and pathnames are written big-endian.

(As another comparison, DNS names such as **www.sheffield.ac.uk** are also written little-endian).

The distinguished name of the topmost entry in the directory (**dc=example,dc=com**, in our example) is called the naming context of the directory, and it's normally based on your organisation's DNS name (**example.com**) because this is guaranteed to be unique. Setting the naming context to be simply **dc=com** is not appropriate because our directory is not trying to hold data for the entire .com domain!

Each entry in the directory is basically a collection of attributes and values. Shortly, we'll create an entry for a user called **mary**, which includes (among many others) the attributes of:

```
uid: mary
sn: Brown
givenName: Mary
```

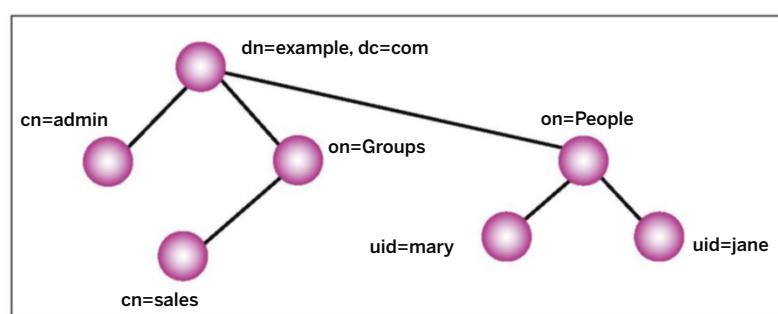
Attributes are a bit like variables in programming languages, where we might say **uid** is a variable with the value **mary**. But don't push the analogy too far because unlike variables, attributes can store multiple values. For example, for **mary** we might see the following

```
telephoneNumber: 01263 987654
telephoneNumber: 07639 123456
```

because the real-life Mary has two phones.

Every entry in the directory must have an attribute called **objectClass**. The value of this specifies a kind of template that specifies which other attributes must be present in the entry, and which may (optionally) be present. So, building towards a more complete entry for our user **mary**, we might see something like this:

```
dn: uid=mary,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
```



» The LDAP Directory Information Tree as developed in the tutorial.

```
» objectClass: shadowAccount
uid: mary
sn: Brown
givenName: Mary
```

Here we see that mary's **objectClass** has three values: **inetOrgPerson**, **posixAccount** and **shadowAccount**. These are effectively combined to determine what attributes **mary**'s entry must have, and which others are optional. Object-oriented programmers might compare these **objectClass** definitions with classes, and indeed object classes can inherit from others, so you'll find that **inetOrgPerson** inherits from **organizationalPerson** which inherits from **person**. These class definitions make up what's known as the schemas of the directory.

But I risk scaring you off with all this theory. In practice, you don't need to get too deep into it if you just want to use LDAP to centralise user account. So let's move on...

The LDAP setup

Our mission this month is simply to set up an LDAP server to store user account information. I'm using Ubuntu 14.04 for this. The principles are the same for any Linux distribution, but the details will differ. I have – after a fashion – already covered this before [see *Dr Brown's Administeria*, p54, LXF187] where I showed how to use *Winbind* to take user account information from Active Directory, which of course has LDAP at its heart. But this time we're going to implement our own LDAP service.

We're about to install the OpenLDAP server, called *slapd*, which will automatically configure itself with minimal user input. However, it takes its naming context (the name of the top-level entry in the directory) from the hostname of the machine, so you should begin by making sure it's included in your **/etc/hosts** file, something like this:

```
127.0.0.1 localhost.example.com localhost
```

```
127.0.1.1 chris-hp250.example.com chris-hp250
```

Now go ahead and install the packages:

```
$ sudo apt-get install ldap-utils
```

This will get you the half-dozen key client tools along with their manual pages. Next, install the server:

```
$ sudo apt-get install slapd
```

You'll be asked to set the password for the LDAP admin account, which is **cn=admin,dc=example,dc=com**. The **slapd** package contains the main LDAP server, and a number of supporting tools along with their man pages, a collection of schemas, and a number of supporting libraries.

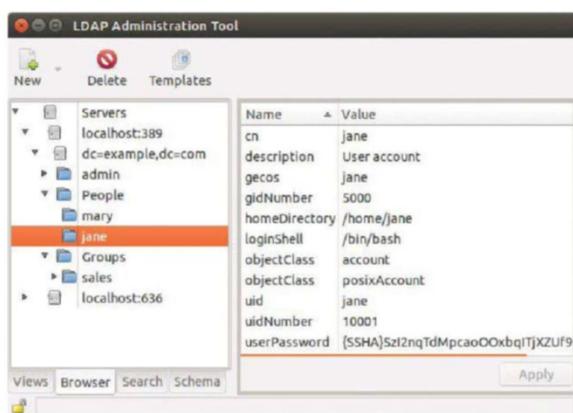
Normally, you'd expect a server to have a config file: **/etc/slapd.conf** perhaps. Although early versions of OpenLDAP did that, the configuration information has now been moved into its own DIT. The LDIF files from which this DIT is loaded are stored in the folder **/etc/ldap/slapd.d/cn=config**. Note however, that you should *not* hand-edit these files. If you need the details, see <http://bit.ly/OpenLDAPAdminGuide>.

Creating a user

As is usual on Debian-derived distributions, installing a service automatically configures the server and brings it into a minimal working state. So we can go right ahead and add some content. Here's the hard way to do it. First, create a file called **populate.ldif** like this:

```
dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
ou: People
dn: ou=Groups,dc=example,dc=com
objectClass: organizationalUnit
ou: Groups
dn: cn=sales,ou=Groups,dc=example,dc=com
objectClass: posixGroup
cn: sales
gidNumber: 5000
dn: uid=mary,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: mary
sn: Brown
givenName: Mary
cn: Mary Brown
displayName: Mary Brown
uidNumber: 10000
gidNumber: 5000
userPassword: marybrown
gecos: Mary Brown
loginShell: /bin/bash
homeDirectory: /home/mary
```

LAT (LDAP Administration Tool) is one of a number of graphical tools for browsing, searching and managing an LDAP directory.



Directories vs databases

Directories such as LDAP and databases such as MySQL both offer a highly structured approach to storing and retrieving data. But they are very different.

First, data in LDAP exists within a tree structure – it's hierarchical. There's no way to have some sort of 'connection' between different branches of the tree. Databases, on the other hand, store information in tables, and can represent foreign key/primary key relationships

between those tables. It's true that an LDAP schema (which defines the attribute types that can appear within a node) is analogous to the schema of a database table (the column names and types), but there's no way you can do a 'join' (in the relational sense) between two pieces of an LDAP directory.

Another distinguishing factor is that directories are designed to be 'read mostly'. Typically, the effort involved in updating an item

in a directory is much greater than the effort of retrieving it. As an extreme case, the addition of a single user in NIS requires the entire password map to be rebuilt. Even more extreme, printed telephone directories are consulted daily by thousands of subscribers. Updating such a directory involves printing and shipping lots of dead trees, and is typically only done once a year. With databases, updates are more frequent and the read/write ratio is more equally balanced.

This is an example of an LDIF file. LDIF is a text-based format that provides an 'external' representation of the contents of an LDAP directory. This file describes two important top-level entries, **People** and **Groups**. Under that, we add a group called **sales** and a user called **mary**. You can see the directory tree we're developing on the image on p57. With this file in place, add it to the directory like this:

```
ldapadd -x -D cn=admin,dc=example,dc=com -W -f populate.ldif
```

You'll be asked to provide the password you set on the LDAP admin account when you installed the server.

Be aware that setting the password for **mary** in this way just stores the plain-text password in the directory – not a good idea. In any case, manually preparing LDIF files is clearly not a convenient way to manage user accounts, so let's have a look at some higher-level tools.

First, the **Idapscripts** package provides a set of shell scripts that wrap around the standard command-line tools to make it easier to manage user accounts and groups. You can install it using:

```
$ sudo apt-get install ldapscripts
```

You'll need to tweak the config file **/etc/ldapscripts/ldapscripts.conf** to reflect the naming context of your directory (**dc=example,dc=com** in our case) and perhaps a few other things. The key things I changed are:

```
SUFFIX="dc=example,dc=com"
```

```
GSUFFIX="ou=Groups"
```

```
USUFFIX="ou=People"
```

```
MSUFFIX="ou=Machines"
```

```
BINDDN="cn=admin,dc=example,dc=com"
```

The last line specifies the user account that I'm going to authenticate against. You'll also need to put the LDAP admin password into the **/etc/ldapscripts/ldapscripts.passwd** file (as defined by the **BINDPWDFILE** parameter in

ldapscripts.conf) like this:

```
sudo echo -n mysecret > /etc/ldapscripts/ldapscripts.passwd
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

The **echo -n** is important; having a new line character at the end of the file will prevent this from working. Now we can add a user to the directory much more easily, like this:

```
ldapadduser jane sales
```

Successfully added user jane to LDAP

Successfully set password for user jane

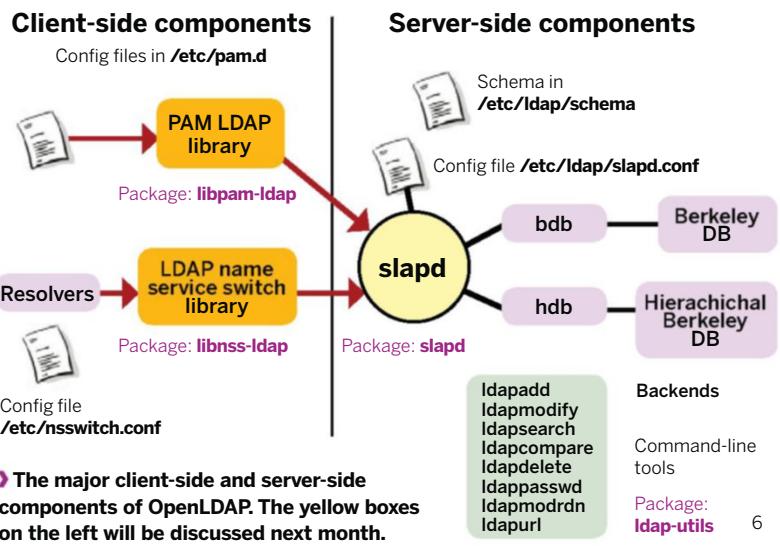
If it doesn't work, look in the log file **/var/log/ldapscripts.log** for hints. The **ldapscripts** package includes several more useful scripts including **Idapaddgroup**, **Idapsetpasswd**,

Idapmodifyuser, and so on. If you just need a command-line solution to managing user accounts in LDAP, these scripts should do the job.

How to look things up

Directories are for looking things up in. From the command line, we use **ldapsearch**. Here, we look up **jane**'s numeric UID:

```
$ ldapsearch -x -LLL -b dc=example,dc=com 'uid=jane'
uidNumber
```



```
dn: uid=jane,ou=People,dc=example,dc=com
```

```
uidNumber: 10001
```

Here, we're starting our search at **dc=example,dc=com** (the top of our directory tree) and we're searching for entries that have an attribute called **uid** with the value **jane**. For each matching entry (there will actually only be one) we print out just the **uidNumber** attribute. We can print out several attributes if we want, like so:

```
$ ldapsearch -x -LLL -b dc=example,dc=com 'uid=jane'
uidNumber loginShell
```

or we can display all the attributes of the matching entry:

```
$ ldapsearch -x -LLL -b dc=example,dc=com 'uid=jane'
```

Point and click

If you'd prefer a graphical tool, there are several that will let you explore and manage an LDAP directory. One such is **LAT** (LDAP Administration Tool) which is in the Ubuntu repositories, so installing it is a breeze:

```
sudo apt-get install lat
```

Notice, though, that it's a .NET application so installing it will also drag in the Mono runtime if you don't already have it. **LAT** is a great little tool for managing users and groups, exploring and searching the directory and examining the schemas and I would strongly recommend you spend some time playing with it.

Other graphical tools worth looking at include **phpLDAPadmin** (web-based) and **Apache Directory Studio**.

That concludes the first part of this tutorial. We now have an LDAP directory up and running that we can use as centralised identity store to authenticate against. Next month, I'll look at the client-side changes needed to actually use it. I also plan to look at the business of replicating the directory to avoid a single point of failure and/or to distribute the load. We'll also see how to set up an encrypted connection to the server using SASL. LXF

A lightweight protocol?

If you look at the complexities of the LDAP protocol you may end up wondering why it's called lightweight. Lightweight compared to what, exactly? Well, LDAP's roots are in the X.500

directory service, a much more complex framework dating from 1988 that was originally intended to run over the OSI network protocol stack, which was a seven-layer model that was being heavily

promoted at the time, until sanity won the day and TCP/IP prevailed. LDAP was originally conceived as a lightweight desktop protocol that could gateway requests to X.500 servers.



The best new open source software on the planet



Alexander Tolstoy

hops from node to node like a cyber mountain goat, scaling the heights of Mount Interweb in search of the sweetest open-source software.

LXF HotPicks

CherryTree » Lunar Linux » OpenSSH » Snapper » Remmina » Wal Commander
GitHub Edition » rTorrent » Neverball » O A.D. » Konversation » Tomahawk

Note-taking tool

CherryTree

Version: 0.35.3 **Web:** www.giuspen.com/cherrytree

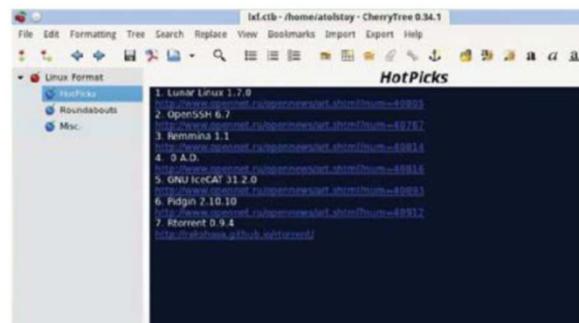
If you're fond of storing notes as plain text files, you know that at some point the whole mess will force you either to migrate to a dedicated note-taking application, or to organise your files into some kind of structure, applying rules to file naming and where you put sub-folders. There is, however, an alternative: *CherryTree*.

A GTK text editor, *CherryTree* insists that you define nodes and sub-nodes in your data tree first; only after that do you populate those nodes with content. You can envision the *CherryTree* document as the root of a tree, and each of its nodes as a branch in that tree, with sub-nodes as sub-branches.

CherryTree supports rich text formatting, syntax highlighting, links, images and even simple tables. Of course, it isn't as powerful as a fully featured word processor, but it doesn't have to be: it's your internal tool for creating sophisticated data structures primarily for use on your desktop.

You can store your data either as a XML file or a SQLite database. The save dialog even lets you protect your file

“Your internal tool for creating sophisticated data structures.”



➤ **CherryTree looks like a web page composer but feels like a database front-end. There's much to love here.**

Exploring the CherryTree interface

Table of contents

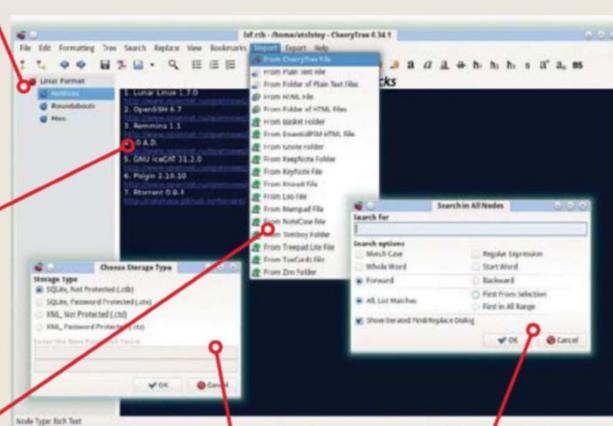
You can use this left pane to navigate through nodes and sub-nodes. You can also create, modify or delete nodes here, or via the context menu, or using the shortcut keys.

Main area

This is where you type and format content. The two default colour schemes are a light background with dark text and vice versa. The spellchecker uses python-enchant.

Import filters

Pretty much every major note-taking application and service is available as an import option – at least the ones with open source credentials.



Storage types

You can select either XML or the SQLite file format. Both can be protected with a password.

Advanced search

Search is a strong point of *CherryTree*. The results can be managed in a separate pop-up window.

with a security password (the file will be encrypted) in any of the file formats.

There are many areas where *CherryTree* excels, but the two most notable are importing and searching. For example, you may already have text files that you want to put to order, or notes taken in other applications such as *Tomboy* or *KeepNote*. *CherryTree* will help you convert them into its more database-like format – just open the Import menu and count how many sources are supported!

After you've migrated the data, you can harness *CherryTree*'s advanced search capabilities. Once everything is stored as a single XML or SQLite file, you can easily find any word or sequence of characters, no matter where it occurs – even regular expressions are supported.

To install *CherryTree*, just look up the standard repository for your distribution; the application is quite well-known and is packaged for dozens of flavours of Linux.

If you decide to compile it from source, be ready to provide Python development packages, as *CherryTree* relies on some Python modules and its GTK2 bindings.

Linux distribution

Lunar Linux

Version: 1.7.0 **Web:** www.lunar-linux.org

It's not always easy to pick a gem out of the treasure chest of Linux distributions available. But what if you want a robust, rapidly growing distro with cutting-edge software and exquisite package manager? Lunar Linux should fit the bill.

You start by downloading an ISO under 400MB in size, which offers a text-based installer for the core system. The install process is quite long-winded – a 14-step wizard – but it is straightforward, covering partitioning your drive, setting mount points; installing the bootloader, creating users, and doing everything else needed to create a bootable Lunar Linux install, which will then go on to offer the tools to build and install applications you might (and surely will) need.

Once booted into your fresh install it's time to enter the Moonbase: the hierarchical software catalogue. This is similar to FreeBSD Ports or Gentoo

Portage. Lunar Linux uses its own unique Lin package manager to install software. In the user's system it sits in **/var/lib/lunar/moonbase**. Software is sorted into sections, and each item is called a 'module'. To install a desktop environment, for example, you just type **lin kde4**, or **lin gnome3**. Package removal is equally as simple: **lrm kde4** will make it go away.

The Lin package manager was created with the aim of removing the "hell of dependency", and it achieves this through quite strict requirements for modules in the Moonbase. Once a new module is approved and tested, it's available for users. Lin uses scripts as well as module metadata to build the

"A rapidly growing distro with an exquisite package manager."



► If you're bored of the branding and flavouring that most Linux vendors bring, try the vanilla Lunar Linux.

module you select, automatically downloading corresponding sources and dependencies. The compilation process may take a lot of time, especially for huge projects like KDE or Chromium, but you get a perfectly robust and flexible system, tailored to your needs. Writing your own modules is a good way to learn the principles of package management, and is perhaps more achievable than with more complex technologies, such as Emerge in Gentoo.

Secure shell

OpenSSH

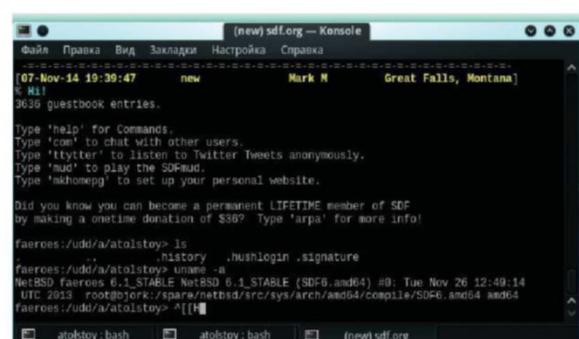
Version: 6.7 **Web:** www.openssh.com

OpenSSH (the OpenBSD Secure Shell) has always been the kind of software that most home users rarely face. But many of us are system administrators in one way or another, even if it's only to access a NAS or portable drive attached to a home router, or remotely connect to your gran's computer. For this, you can use professional administration tools, and OpenSSH is one of them.

However you choose to use it, OpenSSH will provide you with secure and encrypted access to remote systems, either via Secure Shell (SSH) or Secure FTP (SFTP). The primary goal of these security efforts is to prevent eavesdropping, connection hijacking, and other attacks, which can be performed even in your home LAN. OpenSSH has become an essential part of almost any Linux distro, so you are guaranteed to have it.

OpenSSH also keeps on getting better and in version 6.7 it received many enhancements. From the user's perspective, the most thrilling improvement is support for Unix domain socket forwarding: a notable omission in previous versions.

This feature enables a Unix domain socket on the local machine to be forwarded to a remote TCP port, or conversely, a remote TCP port to be forwarded to a local Unix domain socket, in both cases using the same syntax as TCP port forwarding. For example, a remote PostgreSQL database instance could be connected over a secure SSH channel to a Unix



► SDF.org is a public server for playing with SSH tunnels.

domain socket on the local machine by using:

ssh -L/tmp/test.site:mydatabase.net:5432 second_server

It's also possible to connect two local Unix domain sockets over a SSH connection.

Other new features include encryption support using LibreSSL, and resuming interrupted downloads in SFTP; while some legacy code has been dropped, helping to streamline an already very promising application.

If you can't wait for an updated packages for your distro, or you need a custom build, you can also compile OpenSSH from source.

"An essential part of almost any distro that keeps on getting better."

Snapshot tool

Snapper

Version: 0.2.4 **Web:** <http://snapper.io>

Every Linux user would be happy to have a reliable system restore tool: preferably one that's better than what's on offer from the likes of Windows and OS X, but with comparable ease of use.

The may have made you think of ZFS from OpenSolaris, but it's gone dark, and although ZFS is still active (and even ported to Linux), very few people actually use it outside server appliances. Btrfs is a more recent and – more importantly – stable and available filesystem for mainstream Linux. It offers many ZFS-like features, which makes it possible to track filesystem modifications via the filesystem itself, which means that a traditional file backup isn't needed.

Snapper is an open source tool developed by SUSE engineers for managing btrfs snapshots, and it enables you to review changes made to the system and roll them back. *Snapper*

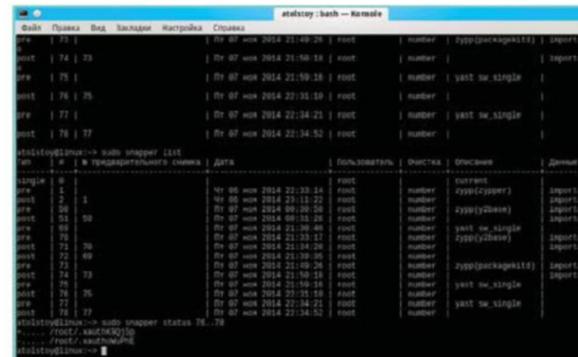
was primarily created for SUSE and OpenSUSE systems, but it's not tied to them; the tool can be compiled for any flavour of Linux. The thing it is tied to is the filesystem that you use.

Even though some experimental ext4 support is present, the main purpose of *Snapper* is to serve as a front-end for the core btrfs features. In order to test it, you're encouraged to have at least one btrfs-formatted partition which you can play with.

Let's assume that you have successfully installed *Snapper*, and that you want to track changes in your **home** directory. For this, you'd use:

```
snapper -c my_home_volume create-config /home'
```

“Snapper enables you to review changes and roll them back.”



► **Snapper can undo changes in your filesystem nearly as easily as using Ctrl+Z in productivity apps.**

As you can see, *Snapper* lets you set a separate volume for any directory, so there can be several volumes and sub-volumes in a partition. To review changes, use **snapper list**, and you'll be provided with a table where every snapshot is recorded. Each has its own number, which is used for manipulating snapshots. For example, to review changes between recent snapshots and roll back to an earlier one, use:

snapper diff 76..78

snapper undochange 76..78

It works like a charm. By the way, many people now consider btrfs to be stable, and *Snapper* provides another reason to switch to it.

Remote desktop client

Remmina

Version: 1.1 **Web:** <http://remmina.sourceforge.net>

Remmina is a tool for accessing remote systems using a graphical user interface. The project is essentially in competition with the proprietary *TeamViewer*.

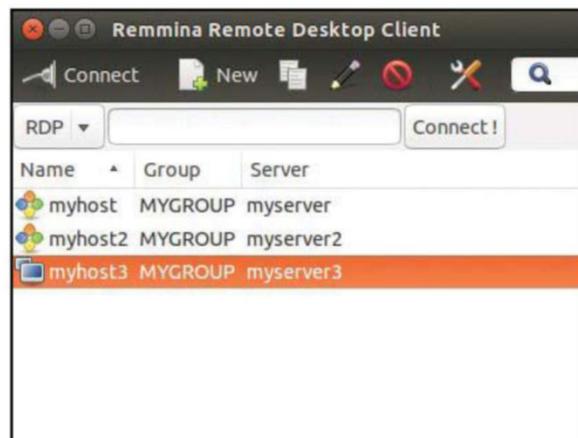
Remmina may remind some people of *KRDC*, and indeed both programs are similar, but *Remmina* fits better in GTK-based desktops (Gnome, Cinnamon and Unity), as it looks cleaner and doesn't bring any KDE dependencies with it.

Remmina is a versatile GTK3 application that includes several connection protocols, such as RDP, SFTP and SSH, and supports the Telepathy and VNC systems. The main *Remmina* application is usually packaged separately from its plugins, so this is the point at which you can customise your installation.

The *Remmina* developers have put noticeable efforts into making the app

more user-friendly. *Remmina* can maintain a list of connection profiles, which are organised by groups; can automatically scroll a remote desktop of a larger size; and also offers a nice floating toolbar over a remote desktop window, with switches and controls.

The original idea behind *Remmina* was to provide a comfortable way of accessing remote computers from an Eee PC. This means a modest screen resolution, so accessing larger desktops leads to painful scrolling and dragging: a problem *Remmina* solves. Not only does it auto-scroll in windowed mode but it also boasts a viewport fullscreen mode, where the remote desktop will



► **Remmina's main window is clean and designed to show many connections in a neat table.**

automatically scroll when the mouse moves past the edge of the screen.

The new 1.1 release is also now compatible with FreeRDP 1.1 (one of *Remmina*'s back-ends) and provides a number of other minor improvements under the hood, such as fixed mouse cursor behaviour.

The wiki page of the project has some guides for building *Remmina* from source in a number of Linux distros, but the process is generally pretty straightforward.

“Automatically scroll a remote desktop of a larger size.”

File manager

Wal Commander

Version: 0.18 **Web:** <http://wcm.linderdaum.com>

Even though the days of *Norton Commander* are long over, many computer users still prefer to use two-panel file managers. Perhaps you've heard of *Midnight Commander*? [See HotPicks, p58, LXF185.] This is the famous Unix clone of Norton's original shell, and it is such a classic that it requires that you run it from the command line. *Wal Commander GitHub Edition* is a little different: it's a true X11 application. (The 'GitHub Edition' part denotes that it was derived from the original *Wal Commander*.)

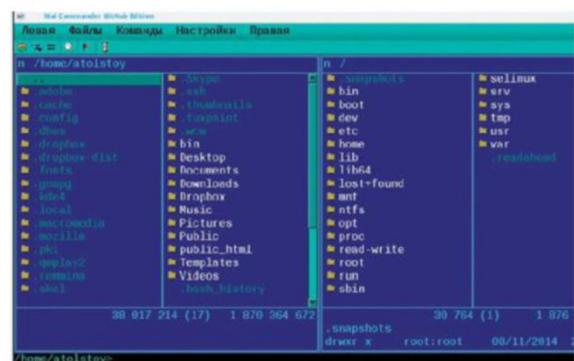
Although it has the classic key bindings (F5 for copying, F8 for deleting, and so on), *Wal Commander* only looks like an old-fashioned double-panel shell. When you start exploring the manager's features, you'll find lots of power tools under the hood. For instance, the default text editor (accessed via F4) supports syntax highlighting for all the languages you

may need (C, C++, sh, Perl, PHP, XML, HTML, Pascal, SQL) and is capable of transcoding characters to another encoding. The command line accepts input both in DOS and Bash syntax.

Wal Commander is also good at communicating, with built-in SMB and FTP/SFTP support, and is perfectly easy to navigate with standard Alt+<letter> shortcuts for controls with highlighted letters.

Generally, *Wal Commander*'s look and feel resembles a Windows application like *Far Manager* and that's true of many of the ways in which it works. For instance, you can set it to show spaces as dots, switch folder icons, control auto-completion and the

"It looks old-fashioned but you'll find power tools under the hood."



► It isn't very pretty, but *Wal Commander GitHub Edition* is a highly usable file manager with lots of modern features.

way hidden files display, and, of course, change appearance settings, including fonts, colours, and buttons.

As its name suggests, *Wal Commander GitHub Edition* can be checked out from its official Git repository, but there are many precompiled binary packages available to download for both Deb and RPM, so you're not forced to compile it from source unless you want to.

The application is monolithic and stores everything in its main binary file **wcm**. Documentation is included in the package, and if you're a first-time user of a two-panel file manager, we advise you to read it first.

Torrent client

rTorrent

Version: 0.9.4 **Web:** <http://rakshasa.github.io/rtorrent>

Last issue, we covered *Geeknote*, a geek utility for managing Evernote [see HotPicks, p62, LXF192]. This time we're focusing on BitTorrent, so we present *rTorrent*: the command-line torrent client. It's based on the libTorrent library, which is developed alongside the client. The client also uses ncurses for showing its pseudo-graphic controls in your shell.

The purpose of *rTorrent* is to give you the full control over your peer-to-peer file sharing, either on remote systems or anywhere without X.org or a graphical web interface.

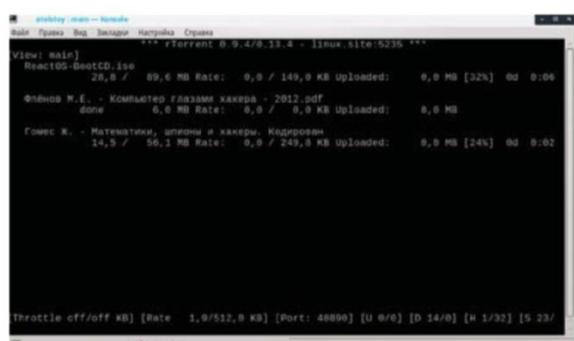
Adding *rTorrent* is extremely easy: it's a tiny application, found in almost all Linux distributions. The tricky part starts when you want to actually use it.

When you run *rTorrent* for the first time, it launches with a complaint about a missing **~/.rtorrent.rc** file. Let's create this by hand. *rTorrent* lets you

explore the inner workings of torrents and specify dozens of core parameters, so it's best to take a dummy (but working) example from <http://bit.ly/1xgtYjg>, which is 117 lines long and provides only the basic settings.

The most obvious next step would be changing the default directories where *rTorrent* looks for new .torrent files and where it puts the downloads. For example, here's a way to schedule *rTorrent* to watch a directory for new torrents, and to stop those that have been deleted:

```
schedule = watch_directory,5,5,load_
start=~/tactive/*.torrent
schedule = tied_directory,6,5,start_tied=
```



► *rTorrent*'s modest black screen is automatically populated with torrents which are found in monitored directories.

```
schedule = untied_directory,7,5,stop_
untied=
```

Other useful options include being able to set the desired minimum and maximum number of peers; to limit the number of downloads and uploads; and to specify the IP address to be shown to the tracker.

rTorrent supports many useful shortcuts, such as lower case a, s and d to gradually increase the upload throttle and their upper case equivalents to increase the download throttle. Ctrl+S starts a download, while Ctrl+D stops it or removes one already stopped. To quit the app, you simply use Ctrl+Q.

"rTorrent lets you explore the inner workings of torrents."

HotGames Entertainment apps

3D platform game

Neverball

Version: 1.6.0 **Web:** <http://neverball.org>

If you've played a classic tabletop Labyrinth game, you'll quickly get the hang of *Neverball*. The game requires you to move a big metal ball by tilting the game world and have it navigate a map full of obstacles within a time limit. Moving platforms and other objects get in your way, making levels progressively harder.

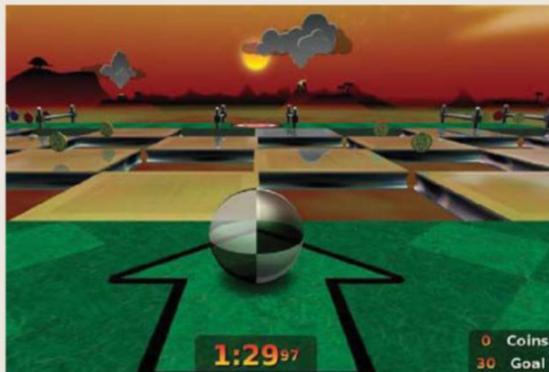
Initially, we suspected that the game was originally for the smartphone, taking advantage of gyro sensors, but it turns out that *Neverball* was first released back in 2003, which means that the game relies on your mouse.

There are 116 main levels in *Neverball* in three course packs, running from easy to hard, as well as 25 extra extremely hard levels. To be honest, we found we could only

complete the 'easy' levels, and then only on a desktop PC, as playing the Android port on a smartphone left us with a pain in our wrists and somewhat mixed feelings – the desktop version is gorgeous, however. As you proceed from level to level, your hand-eye skill develops and you start controlling every tiny move of the ball.

To complete each level you have to collect a set number of coins and reach a yellow-lit exit. Many of the levels are so cool that you have to admire the talent of the level builders: a mix of transparent cubes, loops, mazes,

"The levels are so cool that you have to admire the level builders."



Don't expect *Neverball* to make things easy: each level stresses you out with tricky holes to navigate.

complex curved surfaces and floating platforms, all placed over an abyss and suffering from frequent earthquakes and irresistible pull of gravity.

As well as being addictive, the latest version of *Neverball* also includes better translation packs, Oculus Rift support, and some fancy graphical improvements, such as the transition to the OpenGL ES rendering API and support for high-DPI displays.

Strategy game

O A.D.

Version: Alpha 17 **Web:** <http://playOad.com>

Zero-ey-dee – that's how it's officially pronounced – is a top-quality clone of *Age of Empires II: The Age of Kings*, started back in 2001. In 2003, development moved to a standalone game with its own engine (Pyrogenesis), and in 2010 the game was released as free, open source software.

O.A.D. reconstructs the main civilisations from 500B.C. to 500A.D., with the name standing for the skipped zero year in the middle.

Though it's still officially in alpha, it's a highly playable ancient warfare real-time strategy game. It's also prepackaged for dozens of Linux distributions and is usually presented in two packages: the game engine and a massive 1.5GB data archive. Being curious, we looked inside the **Oad-data** package and found that at

least half of it was occupied by high-quality textures and meshes.

In spite of its hefty footprint, the game starts instantly and shows a brief disclaimer about things missing in the alpha version. Don't worry: all the essential features work fine.

The graphics are smooth and the game is very comfortable to control with smooth scrolling and zooming.

The goal of the game is to develop an ancient city by gathering resources (food, wood, metal, stone, and so on), and constructing buildings and units, both civic and military. As soon as possible, you need to train an army.



O.A.D. is a safe place to victimise nicely rendered trees.

which can even include siege weapons and warships. Training military units costs resources, so the game pushes you towards the only viable strategy: build infrastructure first and your army second.

The time you have for the first stage depends on who you play against. You can choose one or several AI counterparts (but don't choose too many: it slows down the game). If you play against only one enemy (or 'faction') you'll have enough time to build your city in peace before the fighting starts.

"A highly playable ancient warfare real-time strategy game."

IRC client

Konversation

Version: 1.5.1 **Web:** <https://konversation.kde.org>

Internet Relay Chat (IRC) continues to be a popular form of text communication, especially among software developers. *Konversation* is obviously a part of the KDE software stack, and it acts as the 'official' IRC front-end for the KDE team. As a result, it may already be installed in your system, but if it isn't, it's only a few clicks away in your package manager.

To use IRC, you must connect to a server, and then join a channel (equivalent to a chat room) or just engage in private conversations. When you start *Konversation*, it already contains one default server (press F2 for the server list configuration) and a channel there. The vanilla *Konversation* package usually sets Freenode and #kde as a pre-configured server and channel, but some flavours of Linux may customise the default choice (like OpenSUSE, pictured in the image on the right). When you choose to

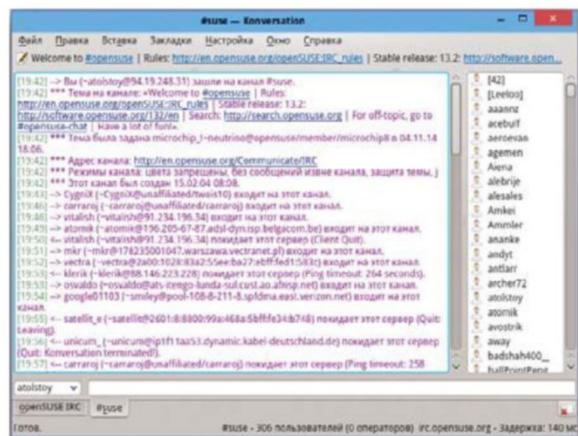
connect to a server, make sure your identity is specified correctly. *Konversation* automatically populates your IRC profile with data, which is derived from your system login details. In you haven't filled these in completely in the past (for example, if you skipped the full name field), *Konversation* will ask you to edit your identity manually.

When you finally join the conversation, you'll see the main chat window and a pane with a list of the users currently connected on the right.

The tabs below the main area list available channels; you can join as many channels at a time as you want.

There are some useful IRC commands that you can use directly in

“Konversation populates your IRC profile automatically.”



► IRC: the convenient way to communicate with Linux users.

the input line. For example, type **/join #channel_name** to join the known channel. If there's no such channel, it will be created and you will be its operator. If you're new to IRC, you can press F1 to consult a very decent *Konversation* handbook for beginners.

Konversation is highly customisable: you can adjust its highlighting feature for better chat readability, set it to play sounds as notifications, and much more. The software is also capable of transferring files and can sync with *KAddressBook*, if you use it.

Media player

Tomahawk

Version: 0.8 **Web:** www.tomahawk-player.org

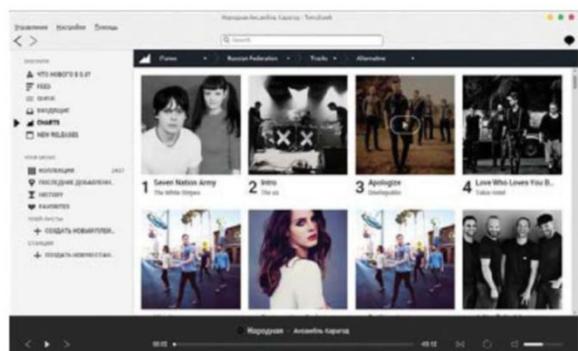
There are so many media players available for Linux that you have to wonder what features can still be missing. *Tomahawk*'s developers think they know and are creating a social media player.

Tomahawk is a modern Qt 5 application with some dependencies from KDE Frameworks 5 and vlc-noX under the hood. The idea of the player is to seamlessly integrate different sources of music into one solid media manager. For that reason, *Tomahawk* is shipped with dozens of plugins, which extend the scope of music you can import. As you would expect, you can connect to Spotify, SoundCloud, Last.fm, Grooveshark, Jamendo, Subsonic, Beats Music and other lesser-known services, but also to cloud storage like OwnCloud and Google Play Music; and play music from other *Tomahawk* users. This means you can

stream shared content either from local buddies in your LAN, or remotely via XMPP (Jabber). These plugins provide a unique experience: one in which it's almost impossible to tell local from streamed content.

When you first launch *Tomahawk*, you're greeted with a setup window, from which you can enable the desired plugins from the first tab, select your local music folder from the second, and change port forwarding and proxy settings from the third.

The *Tomahawk* interface sits somewhere between an email client and a tweaked version of *iTunes*. The Discover section of the sidebar provides



► Tomahawk lets you browse a bewildering range of music sources to find your favourite tunes or discover new ones.

social options, showing you your feed and tracks your buddies have shared with you, and letting you share tracks with them in return (you're supposed to put your suggestions in the queue).

Locally stored music lives just below this, in the Your Music section, where you can surf your collection and add tracks to playlists.

The topmost bar of the player is used for navigating and searching for tracks, while the lower area displays classic control toolbar showing you what's playing now and providing common playback control buttons and volume controls. **LXF**

“Seamlessly integrate different sources of music into one collection.”

Issue 192
Christmas 2014
Product code:
LXFDB0192

In the magazine

More power! Charge up your distro with essential tips and tools. Build a robot and a monstrous 24TB NAS box. Plus: system recovery, Linux certs and our pick of the most productive desktops.



LXF DVD highlights
Ubuntu 14.10 Remix (pick from 5 desktops), ROSA, Rescatux.

Issue 191
December 2014
Product code:
LXFDB0191

In the magazine

Take your Raspberry Pi mastery to the next level with our hot hacks. Learn how to contain everything with Docker and plug in to professional audio production using JACK. Plus: Develop with PHP.



LXF DVD highlights
Hot Pi distros! Kali Linux, Jasper, RetroPie, Pi MusicBox and more.

Issue 190
November 2014
Product code:
LXFDB0190

In the magazine

Origin of the distro – LXF tracks the source of your favourite distro and picks the best of each genus. Plus: we chart Ubuntu's bumpy history as it celebrates 10 years. Also, Pi alternatives and the best web browsers.



LXF DVD highlights
Linux Mint DE, Mageia 4.1, Trisquel 6, CentOS 7 and more.

Issue 189
October 2014
Product code:
LXFDB0189

In the magazine

Discover how to solve your Linux problems with a huge troubleshooting guide. Plus: Chromebook roundup, run Linux on almost any Android device, cryptography explained and Minecraft hacking.



LXF DVD highlights
Tails 1.1 Live DVD, Deepin 2014 and 3 essential rescue distros.

Issue 188
September 2014
Product code:
LXFDB0188

In the magazine

Streamline your desktop, cut boot times and get more from your system with our guide to speeding up Linux. Plus: voice recognition app roundup, the state of education and how good is 4K support in Linux?



LXF DVD highlights
CentOS 7, Arch Linux, Peppermint Five, Rescatux 0.32, and more.

Issue 187
Summer 2014
Product code:
LXFDB0187

In the magazine

Use our foolproof guide to switch from Windows, including backup and dual-booting. Plus: turn a Raspberry Pi into a wiki server, choose an HD media player, run Windows apps in Linux with Wine, and more.



LXF DVD highlights
Ubuntu 14.04 LTS, Mint 17, SystemRescueCd and more.

To order, visit www.myfavouritemagazines.co.uk

Select Computer from the all Magazines list and then select Linux Format.

Or call the back issues hotline on **0844 848 2852**
or **+44 1604 251045** 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.

★ SAVE UP TO 57% ★

- On the perfect gift this Christmas -

WITH OUR VARIED SELECTION OF TITLES YOU'LL FIND THE PERFECT PRESENT FOR THAT SPECIAL SOMEONE



SAVE UP TO 45%
FROM £25.49



SAVE UP TO 45%
FROM £23.49



SAVE UP TO 40%
FROM £25.49



SAVE UP TO 70%
FROM £15.99



SAVE UP TO 50%
FROM £23.49



SAVE UP TO 40%
FROM £26.49



SAVE UP TO 55%
FROM £12.99



SAVE UP TO 40%
FROM £17.99



SAVE UP TO 35%
FROM £22.49



SAVE UP TO 50%
FROM £20.99



SAVE UP TO 50%
FROM £21.49



SAVE UP TO 40%
FROM £25.49

- Save up to 57% off the cover price
- Delivery included in the price
- Over 45 magazines covering cars, film, music, technology, gaming and more
- Free gift card you can personalise to announce your gift
- Huge range of items priced under £20
- PLUS! 20% off overseas subscriptions



2 easy ways to order



myfavouritemagazines.co.uk/Z501



Or call us on
quote Z501

Lines open Mon to Fri 8am – 9.30pm
and Sat 8am – 4pm

Savings compared to buying 2 year's worth of full priced issues from UK newsstand. This offer is for new print subscribers only. You will receive 13 issues in a year. Full details of the Direct Debit guarantee are available upon request. If you are dissatisfied in any way you can write to us or call us to cancel your subscription at any time and we will refund you for all unmailed issues. Prices correct at point of print and subject to change. For full terms and conditions please visit: myfavm.ag/magterms Offer ends: 31st January 2015

» **HDDs** Erase your personal data from drives and protect your confidentiality

HDDs: Erase drives securely

Sean Conway explores the options for securely erasing confidential data from old hard drives – from *DBAN* and Secure Erase to a Lee-Enfield rifle!



Our expert

Sean Conway is a semi-pitiless learner with experience that spans vacuum tubes and complementary metal-oxide semiconductors.

So you're thinking of donating or selling a few used computers. You realise that it's important when repatriating PCs to sanitise the media to protect the confidentiality of your personal data – so what do you do? According to the National Institute of Standards and Technology, there are three levels of sanitisation: clearing, purging and destroying.

Clearing prevents data from being recovered by data-, disk- or file-recovery utilities. Purging prevents laboratory technicians from using signal-processing equipment to recover the data, and destroying data is... well, destroying the physical media so that it cannot be reused at all.

For this tutorial, we tested seven different methods for erasing a 160GB 5400.6 Seagate laptop hard drive: the standard **rm**, **format** and **shred** commands; *DBAN* software and Secure Erase firmware; degaussing in a specialist machine; and physical destruction. Not every reader has access to equipment worth \$10,000 or wishes to put hard drives at risk to discover the outcomes, so you, dear reader, get the benefit without the risk.

You can read more about how we carried out the tests on the opposite page (see *Our testing procedure*). The results are summarised in the following table.

As you can see, five of the methods provide some measure of sanitisation. Which one you adopt depends largely on use case. The **shred** command, for instance, does make data files unreadable, but it has to be run manually on individual files, which makes it impractical for sanitising an entire hard drive. At the opposite end of the scale, degaussing and destruction render the hard drive unreadable, so don't use them if you want it to be usable by its new owner.

In the tutorial, we will show you how to employ the final four methods in the table for yourselves. (We'll assume that you can manage to **shred** data on your own.)

A word of warning: don't explore the techniques in this tutorial on your personal PC. The tools we have used have the ability to render a hard drive unreadable. If you're going to follow along, use a separate computer with a hard drive that can be sacrificed in the name of learning.

Method	Sanitation type	Recovery result
rm command	Below minimum	3 files readable
format command	Below minimum	3 files readable
shred command	Purging: individual files only	0 files readable
<i>DBAN</i>	Purging: maximum time	0 files readable
Secure Erase	Purging: minimum time	0 files readable
Degaussing	Purging: special hardware	0 files readable
Destruction	Destroying	0 files readable

Deleting data with DBAN

DBAN (*Darik's Boot And Nuke*, <http://dban.org>) is a downloadable application distributed by Blancco. The free CD image the vendor provides is designed for the home user to erase hard drives, but the company also has a licensed commercial product for business applications. The tool is a collection of wiping algorithms and configurations a user selects to delete the contents of a targeted hard drive.

When fired up, the software will overwrite the hard drive with random or pre-selected data. The process overwrites the hard drive locations for file data, file system data and all addressable locations on the hard drive. Its objective is to replace all the data on the drive with random data generated by an algorithm. *DBAN* can only be used on functioning hard drives: drives detected by BIOS and good for service.

Download *DBAN* 2.2.8 from the website and create a bootable CD/DVD. When the software boots up, select the *DBAN* option from the list. You can select any drive detected. Make changes to the software using the shortcut keys listed at the bottom of the screen.

Select the disk you want to sanitise and press F10 to initiate erasure. The Statistics sections of the screen will become populated once the process starts. It can take a long time to complete because *DBAN* has to generate and write data to every storage bit on the hard drive.

Using Secure Erase

Alternatively, hard drives manufactured after 2001 come with

Quick tip

If you're not risk adverse, Fedora and Ubuntu variants have a *hdparm* tool set that can be installed with a *yum* and *apt-get* package management tool respectively.

Our testing procedure

Before we tested each different method for sanitising a hard drive, we put the drive through a standard preparation process. First, we purged it of data using RCMP TSSIT OPS-II data-wiping software on the *DBAN* CD. This method has been deprecated by the Canadian government and replaced by CSEC ITSG-06, a method that takes advantage of Secure Erase. The RCMP method was chosen not because it's superior, but because it had a special Red Serge appeal for the author of the tutorial, who is Canadian.

After the data was purged, we partitioned the drive (using **fdisk**), formatted to ext3 (using **mkfs.ext3**) and copied files of nine different types (DOC, DOCX, EPUB, JPG, PNG, ODS, ODT, TXT and ZIP) to the drive, readying it for testing.

After we had completed each different method of sanitisation, we used the *PhotoRec* file-recovery application demonstrated in a previous issue (see *Tutorials*, p68, **LXF190**) to attempt file recovery.

firmware that provides sanitisation capabilities without needing additional software. An ATA drive detected by system BIOS can be sanitised by issuing a Secure Erase command from a terminal window. Any commands issued will be executed directly on the hard drive.

During this process, the hard drive can be locked, making it inaccessible to the OS. During testing we managed to do this unintentionally, rendering the drive useless. (Fortunately, we were able to restore the drive using the user-defined password security command.)

For this reason, the risks associated with having easy access to Secure Erase can be mitigated by using a separate OS specifically for testing purposes: something like the bootable SystemRescueCd (www.sysresccd.org), which contains the Secure Erase command set.

The SystemRescueCd image is a collection of tools for administering or repairing Linux or Windows computers. The software boots from a CD/DVD or USB with no installation required, and the selection menu on boot up enables the user to work in the CLI or a GUI.

The site has abundant documentation that explains the process of creating a bootable CD, DVD or USB drive. Once you've booted your preferred SystemRescueCd media, you can select support for a GUI interface at the first menu. You will be presented with root access to a full Linux OS and supporting applications. If the system is plugged into a network, part of the boot process will request an IP address via DHCP to establish access to the network and, hopefully, the internet.

In the code for this tutorial, you will see two parameters: **X** and **#**. **X** is the device ID. This information can be collected from an application like *GParted* or scraped from the dmesg log. **#** is the host number. We'll determine this later.

Open a terminal. You will now use the **hdparm** command (see *Quick tip*, p68), as follows:

```
hdparm -I /dev/sdX
```

In the output, you should see something like this:

Security:

Master password revision code = 65534

supported

```
[scorway@fedoral9-inane-ca ~] $ sudo fdisk -l /dev/sdc
Disk /dev/sdc: 160.0 GB, 160041885696 bytes, 312501808 sectors
Units = sectors of 1 = 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x312121e6

Device Boot Start End Blocks Id System
/dev/sdc1 2048 312501807 156209880 83 Linux
[scorway@fedoral9-inane-ca ~] $ sudo mkfs.ext3 /dev/sdc1
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label:
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks. Stripe width=0 blocks
9773056 inodes, 39072470 blocks
1953623 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
1193 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 29376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872
```

» **O Canada! Our home and native land! Secure Erase in all thy sons' command.**

not	enabled
not	locked
	frozen
not	expired: security count
supported	enhanced erase

The parameters **not frozen**, **not locked** and **enabled** must be established on the hard drive in order to proceed with a Secure Erase. The wiki (<http://bit.ly/SecureErase>) includes an explanation of why the conditions may exist and how to correct them.

In our tests, we had success in establishing the **not frozen** parameter using a process set out by Edoardo Liverani (<http://bit.ly/Liverani>), so that's the one we'll cover here. Note: the hard drives we used for our tests supported hot swapping (that is, they could be removed while computer is powered up). It's a very bad idea to remove a standard drive or install components while your system is powered.

First, you'll need to determine the host number:

```
ls -ld /sys/block/sdX
```

In the output this returns, look for a text string like **host5**. This is the host number you need. Detach the drive and power off. (Remember what we said about not hot swapping standard drives.) Now run:



» **High-tech disk sanitisation via DBAN, recorded using a far lower-tech camera.**

» **For print and digital subscriptions** See www.myfavouritemagazines.co.uk

Tutorial Erase HDDs

Quick tip

All the Secure Erase documentation warns about the risks. Some of the commands even require the attribute `--i-know-what-i-am-doing` before they will complete.

- » `echo 1 > /sys/block/sdX/device/delete`
Reattach the hard drive, and run:
`echo " - - -" > /sys/class/scsi_host/host#/scan`
Check the drive status with **hdparm** to ensure the **not frozen** parameter is established.
To proceed with a Secure Erase, the next security parameter must be changed to **enabled**. To enable security on the hard drive, enter the following at the command line:
`hdparm --user-master u --security-set-pass sean /dev/sdX`
You can use any password you like in place of **sean**. If you want to set the drive back to **not enabled**, use:
`hdparm --user-master u --security-disable sean /dev/sdX`
To remove the **locked** security option:
`hdparm --user-master u --security-unlock sean /dev/sdX`
You can initiate a Secure Erase as follows:
`hdparm --user-master u --security-erase sean /dev/sdX`
Before returning the hard drive to service, ensure the **not enabled**, **not locked** and **not frozen** parameters are re-established or the OS will not be able to access the drive.

Degaussing and destruction

Let's cover two final methods of sanitisation to wrap up our discussion. Subjecting a hard drive to a degausser or physical destruction through pulverisation or shredding will, quite



» **The Terminator of data wiping: the HD-3WXL Data Eliminator.**

DBAN and UNetbootin

If you prefer to use *UNetbootin* in a Linux distro to create a bootable USB thumb drive from the downloadable *DBAN* image, some file editing is required for the USB to function.

By default, the bootable USB created with *UNetbootin* stalls and generates a RAM disk error message – which is just as well, since the configuration created by *UNetbootin* autonukes a hard disk on boot-up without requiring user input.

On his blog (<http://bit.ly/UNetbootinBugs>), Alex Pounds has

supplied a workaround, which we've paraphrased below.

In the **syslinux.cfg** file on the USB thumb drive created using *UNetbootin*, make the following text changes:

- » Replace all occurrences of **ubninit** with **ISOLINUX.BIN**.
- » Replace all occurrences of **ubnkern** with **DBAN.BZI**.
- » Delete **--autonuke** from the line in **label unetbootindefault**.

Don't forget to save the changes and remember that the entries are case sensitive.

obviously, render the data and the hard drive unusable.

But how do these methods actually work?

Hard drives contain circular metal or glass platters coated with iron oxide. Each platter spins between metal coils called recording heads. When the coil head receives electrical signals, the head magnetises the surface of the platter. Data is recorded on the platter as a series of magnetic 1s and 0s.

A hard drive degausser is an electronic device that generates an intense magnetic field. This magnetic field will reset (erase) media so that it can't be reused. Placing a hard drive in the magnetic field of a degausser scrambles the magnetic information contained on the platters.

We used a Garner HD-3WXL degausser to sanitise our test hard drive. (You can see a video recording of the process at <http://bit.ly/Degausser>). The HD-3WXL is a capacitor discharge degausser. The internal electronics charges capacitors to store energy, like charging a battery. When the capacitors are discharged through special coils in the device, a large electromagnetic pulse (EMP) is produced. The magnetic field is so intense it randomises the magnetic material in the hard drive. The EMP produced in a degausser is not unlike the one that's produced in a nuclear explosion.

The degaussing process not only removes all the user data, but also the servo track information laid down on the platters during the hard drive factory build. Without this information the drive will no longer function, so a degaussed hard drive won't boot up.

After we had degaussed the drive, we installed it in our test computer and tried to boot it. The BIOS detected a hard drive, but then failed on analysis, reporting an error on the drive or drive firmware before it continued the boot cycle.

Extreme measures

Our final sanitisation method was to physically destroy the hard drive. This means breaking the platters of the drive into pieces to make reading data from the media physically impossible. This can be done using specialist mechanical shredders or a mechanical pulveriser. Security Engineered Machinery, which makes this kind of equipment, has some cool videos of hard drive destruction: <http://bit.ly/SemShredHDDestruction>.

If you don't have a shredder, you can use a power drill to make holes in the platters or scratch their surfaces. It has been suggested bending the platters may be sufficient to prevent reading the data. However, some theorists speculate that government organisations have technology that can recover data unless the platter pieces are fully destroyed.

To finish the article the author took advantage of a destruction method that's not entirely recommended, especially if you live in the UK. But in Canada it's not so much of a problem.

There's a full video on this month's **LXFDVD** of a bullet from a Lee-Enfield bolt-action, magazine-fed, repeating rifle shattering the tutorial hard drive. We would have used a baseball bat and recreated that scene from *Office Space*, but we weren't convinced it would do enough damage.

So there you have it, system administrators: a range of different sanitisation methods to ensure that hard drives you pass on don't contain personal data. Hopefully you find one that meets your requirements, and get rid of some of those old computers. **LXF**

» **Never miss another issue** Subscribe to the #1 source for Linux on page 32.

ipset: Simplify your firewalls

Daniil Baturin demonstrates how to take the pain out of maintaining a firewall by using ipset to make configurations easier to read and modify.



Our expert

Daniil Baturin is a programmer and a network administrator, and the maintainer of the VyOS router distro (<http://vyos.net>) and a few other FOSS software projects.



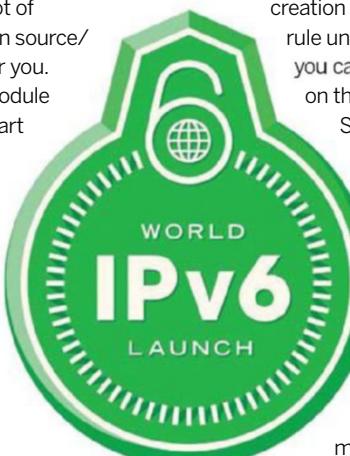
Ipset is a Netfilter extension that enables you to create lists of IPv4 and IPv6 addresses, networks and TCP/UDP port numbers and use those lists in iptables/ip6tables rule source and destination options. In complex firewall setups it can make configurations easier to read and modify. If your firewall config contains a lot of near-identical rules with slight variations in source/destination addresses or ports, ipset is for you.

Ipset consists of two parts: a kernel module and a userspace admin tool. The kernel part is now in the mainline kernel, and the tool is usually available from repos. Some distros also include service wrappers for loading ipset configs on boot, such as ipset-service in Fedora.

Computer networks are getting more and more complex, and people are coming to rely on them more than ever. Firewalls still remain the first line of defence, which means that firewall policy complexity is growing as well.

Quick tip

Use the **family inet** option with lists that include IPv4 addresses. For lists that include IPv6 addresses, use **family inet6**. If you don't specify a family it's assumed to be IPv4.



Sometimes system administrators may get a request such as 'allow HTTP to host 192.0.2.1'; but more often requests are more general, like 'allow SSH from all development workstation', or 'allow HTTP and HTTPS from all office computers', or 'allow SMTP, IMAP and something else from these remote sites'.

You may also need to allow or deny services that use more than one protocol and port, such as IPsec, which uses a couple of IP protocols for data and UDP for key exchange; or SIP, which may use either TCP or UDP for signalling and also needs a range of UDP ports for media; or even Active Directory, which needs almost a dozen TCP and UDP ports.

Of course, you can do all of these with iptables alone. The obvious problem is that complex requirements often need more than one rule to fulfil. Sometimes you can cope with protocol complexity by matching the **RELATED** state and using conntrack modules; sometimes you can cope with multiple addresses and ports by writing scripts or using ready-made ones like those in the *Shorewall* configuration tool. Scripts that generate iptables rules don't remove the complexity though, they just move it to elsewhere, so your **iptables -L** output stays just as long as if you wrote the rules by hand (or sometimes longer). If you're finding this type of complexity becoming a problem, then ipset is one solution.

The workflow

Generally, an ipset workflow runs like this: create a set (list), add some entries to it and create an iptables/ip6tables rule that refers to it.

So far, so simple. Lists can be different types, so you can't add entries to a list until you create it and specify the type at creation time. You can't reference a list in an iptables rule until you create that list. The good thing is that you can modify a list referenced in an iptables rule on the fly without reloading any iptables rules.

Suppose, then, that we have the following requirements:

- » Allow SMTP, IMAP, and POP3 from trusted networks 192.0.2.0/24 and 2001:db8::/64
- » Allow SSH from hosts 192.0.2.10, 192.0.2.15, 2001:db8::100, 2001:db8::105
- » Allow port 5000 from host 203.0.113.5, port 5010 from host 203.0.113.10, and port 5020 from 203.0.113.42

Now we'll see how we can cut down the number of rules and make this setup more manageable. First, we'll deal with the email part. »

Tutorial ipset

» As you can see, we have more ports than network addresses here, so we'll go ahead and create a list of ports and reference it in rules for those subnets like so:

```
# ipset create EmailPorts bitmap:port range 0-65535
comment
# ipset add EmailPorts 25 comment SMTP
# ipset add EmailPorts 110 comment POP3
# ipset add EmailPorts 143 comment IMAP
```

In the **create** command, **EmailPorts** is the name of the list. Type **bitmap:port** is the type you need to store ports. The option for **port range** is mandatory, but you can make the range more limited if you want. The ipset extension for storing comments along with the entries is optional, so you need to explicitly enable it with the **comment** option.

Now you can make sure everything is right by viewing the set we've just created using the following command:

```
root@eirin ~]#
[root@eirin ~]# ipset create EmailPorts bitmap:port range 0-65535 comment
[root@eirin ~]# ipset add EmailPorts 25 comment SMTP
[root@eirin ~]# ipset add EmailPorts 110 comment POP3
[root@eirin ~]# ipset add EmailPorts 143 comment IMAP
[root@eirin ~]#
[root@eirin ~]# ipset list
Name: EmailPorts
Type: bitmap:port
Revision: 2
Header: range 0-65535 comment
Size in memory: 532596
References: 0
Members:
25 comment "SMTP"
110 comment "POP3"
143 comment "IMAP"
[root@eirin ~]#
```

» It's a good idea to assign memorable names to your ports, so that you know what each one is used for a month down the line.

ipset list EmailPorts

You can also view all configured lists with **ipset list** with no arguments. Now when we have a port list, we can reference it in Netfilter rules:

```
# iptables -A INPUT -s 192.0.2.0/24 -p tcp -m set --match-set EmailPorts dst -j ACCEPT
# ip6tables -A INPUT -s 2001:db8::/64 -p tcp -m set --match-set EmailPorts dst -j ACCEPT
```

Note that you can't specify a protocol along with the port in this type of list; what you need to do is specify the protocol in your firewall rule. This works with TCP and UDP (or even SCTP) alike. The option **-m set --match-set** is where you reference your ipset object. It consists of two parts: the list and the direction. The list requirement is pretty obvious: it's just your list name. The direction must be **src** (source) or **dst** (destination – later we will see that there can be more than one direction). In this case we are interested in destination ports, so we use **dst**.

Port lists can be used in iptables and ip6tables rules alike; there is no need to adjust anything for either protocol. If you need to add a whole range of ports to a list, there's a shorthand for this, as follows:

```
# ipset add PortList 15000-16000
```

The downside is exactly that: it's a shorthand, and ipset will actually add all ports from the range to the list, which can seriously impair readability, so for huge ranges it may be better to just specify them in the iptables/ip6tables rule.

A list of hosts

Now to the SSH requirement. In this case we have multiple hosts and only one port, so it's reasonable to create a list of hosts and then reference them in a rule for SSH. In ipset, you cannot mix IPv4 and IPv6 addresses in one list, just like you have to use iptables or ip6tables for different protocols. So we'll have to create two lists:

```
# ipset create TrustedHosts hash:ip family inet comment
```

IPv6 vs IPv4

In terms of forwarding, filtering and policing, IPv6 is not really all that different from IPv4. The most noticeable difference is that the **iptables** command is called **ip6tables** instead, but all the options, apart from some really protocol-specific ones, are the same. Iptset options are no exception: you can use the same format with both **iptables** and **ip6tables** and there are no tricks to remember.

Other options have differently named counterparts in IPv6: the time to live field is called TTL (**-m ttl -ttl-[eq|lt|gt]**) in IPv4 and HL (which stands for 'hop

limit') in IPv6. The IPv6 equivalent is **-m hl --hl-[eq|lt|gt]**. Another example is the ICMP protocol, which is slightly different for IPv6. The option for ICMP messages is **-m icmpv6 --icmpv6-type=<type>**.

Of course, there are certain options that have no IPv4 counterparts. Those include the Mobility Header used in mobile IPv6, Destination Options, the Hop-by-Hop Options header, and a few more. However, these are nowhere as widely used as options like source and destination address. None of your iptables knowledge is lost in transition to IPv6.



» As the internet gradually switches over to IPv6, now is the time to start learning about the protocol. Fortunately, many options are similar to IPv4.

» If you missed last issue Call 0844 848 2852 or +44 1604 251045

```
# ipset add SSH_Hosts 192.0.2.10 comment "Alice's
workstation"
# ipset add SSH_Hosts 192.0.2.15 comment "Bob's
workstation"

Here hash:ip is the type for storing IPv4 or IPv6 host
addresses, and family inet is the IP address family specifier.
If the address family isn't specified, it's assumed to be IPv4,
so in this case that option is redundant. Note: this is a
mandatory requirement for IPv6.
```

```
# ipset create TrustedHosts6 hash:ip family inet6 comment
# ipset add TrustedHosts6 2001:db8::100 comment "Alice's
workstation IPv6"
# ipset add TrustedHosts6 2001:db8::105 comment "Bob's
workstation IPv6"
```

The only thing left to do is set up the Netfilter rules:

```
# iptables -A INPUT -p tcp --dport 22 -m set --match-set
TrustedHosts src -j ACCEPT
# ip6tables -A INPUT -p tcp --dport 22 -m set --match-set
TrustedHosts6 src -j ACCEPT
```

It's possible to use multiple **--match-set** options in one rule. If we were to allow email from these hosts, we could also reuse the **EmailPorts** list from earlier and do something like this:

```
# iptables -A INPUT -p tcp -m set --match-set EmailPorts dst
-m set --match-set TrustedHosts src -j ACCEPT
```

There is also a type for lists of networks, **hash:net**, which holds subnet addresses:

```
# ipset create NetworkList hash:net
# ipset add NetworkList 10.1.0.0/24
```

IPs and ports

Now we will look at a more complex example: a list of pairs rather than individual objects. In this case, it's a list of pairs of IP addresses and port numbers, requiring multiple ports and addresses. This might seem a bit artificial, but I've been in situations where seemingly random addresses and ports were in fact related to each other: for example, when I had to

allow access to an application running on NAT-ed internal machines to support technicians from a company that was using multiple offices.

Ipset supports pairs (and even triplets) of hosts and ports or networks and ports. The type for address and port pairs is **hash:ip,port**:

```
# ipset create AppSupport hash:ip,port
# ipset add AppSupport 203.0.113.5,tcp:5000
# ipset add AppSupport 203.0.113.10,tcp:5010
# ipset add AppSupport 203.0.113.42,tcp:5020
```

In this case we need to specify two directions in the **--match-set** option: first for the address and then for the port entry:

```
# iptables -A INPUT -m set --match-set AppSupport src,dst
-j ACCEPT
```

You can specify any combination of **src** and **dst**, such as **dst,src** or **src,src** – whichever suits your particular purpose. Lists of IP/port and network/port pairs also allow you to specify the protocol along with the port, which can save you some time and effort when dealing with protocols that can use both TCP and UDP, such as DNS or SIP. If we were to allow DNS queries and zone transfer from some hosts, we could use a list like this:

```
# ipset create DNS hash:ip,port
# ipset add DNS 192.0.2.200,udp:53
# ipset add DNS 192.0.2.200,tcp:53
# iptables -A FORWARD -m set --match-set DNS dst,dst
```

Just like iptables, ipset enables you to load the rules from a file and output those rules in format suitable for loading, as follows:

```
# ipset save > /path/to/ipset.save
# ipset restore < /path/to/ipset.save
```

Hopefully ipset will help you keep your firewall configuration short, expressive, and much easier for you to maintain. If you need more information, you can visit the project website, which is <http://ipset.netfilter.org>, and read the man pages included in the package. LXF

```
[root@jbmachine ~]# ipset create TrustedHosts hash:ip family inet
[root@jbmachine ~]# ipset add TrustedHosts 192.0.2.10
[root@jbmachine ~]# ipset add TrustedHosts 203.0.113.14
[root@jbmachine ~]#
[root@jbmachine ~]# ipset create TrustedHosts6 hash:ip family inet6
[root@jbmachine ~]# ipset add TrustedHosts6 2001:db8:ff::100
[root@jbmachine ~]# ipset add TrustedHosts6 2001:db8:ab:ac::201
[root@jbmachine ~]#
[root@jbmachine ~]# iptables -I INPUT -m set --match-set TrustedHosts src -j ACCEPT
[root@jbmachine ~]# ip6tables -I INPUT -m set --match-set TrustedHosts6 src -j ACCEPT
[root@jbmachine ~]#
[root@jbmachine ~]# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT    all  --  anywhere             anywhere            match-set TrustedHosts src
[root@jbmachine ~]# ip6tables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT    all  --  anywhere             anywhere            match-set TrustedHosts6 src
```

You can create groups of trusted or not-so trusted hosts easily enough for both IPv4 and IPv6 addresses.

» **Never miss another issue** Subscribe to the #1 source for Linux on page 32.

Minix 3: Test a microkernel OS

Richard Smedley finds Minix 3 has gone way beyond its educational origins with ARM compatibility, but microkernel reliability is still at its heart.



Our expert

Richard Smedley has tried all the POSIX compliant OSes available from the '90s, from V2_OS via BeOS to AtheOS, and even GNU Hurd.



Minix 3 is a microkernel-based Unix clone, targeted towards robustness, reliability, and a small memory footprint. In a microkernel, drivers and servers are isolated, running in User Mode, and restricted in the effects they can have on the rest of the system (more on this later). The price for this is a small performance overhead, and some design challenges, but performance is not as important here as security and stability for most users.

Microkernels have a long history. In the '80s, academics were convinced they were the only way forward for OS development, and Minix author Andrew Tanenbaum never considered designing Minix in any other way. It also meant Richard Stallman turned to a microkernel as the basis of the Hurd, the kernel of the GNU OS. The imminent arrival of GNU – and BSD Unix – motivated Tanenbaum to keep Minix as an academic OS, but that fitted well with the aims of having something small enough to teach from, and cover in his famous book, *Operating Systems: Design and Implementation*.

Legal problems, and then technical challenges, held up the development of the Hurd, but Minix's tight focus enabled Tanenbaum to produce Minix 1 more or less single-handed, and three decades of putting the few thousands of lines of kernel code under close scrutiny while his doctoral students at Vrije Universiteit, Amsterdam worked on every area, have built a solid foundation for Minix 3.

Despite historical spats with Linux (see *Ancient history*, p75) and the education-targeted early releases, Minix has become an appealing OS for many users, particularly as it's gradually integrated the NetBSD userland. This has been achieved largely because of millions in EU funding focused on 'secure and reliable' OS development, and many generous contributions from the Google Summer of Code.

The Minix team presented the latest version at FOSDEM this year and the 'MINi UNIX' has grown in scope, but not in bloat. ARM support, for instance, has made it into the master code branch, and the team say that version 3 has "more complete source compatibility with NetBSD in terms of utilities, calls, types (lots of 64-bit), toolchain, codebase and packages", with "all Minix-specific code in a top-level `minix/` subdir".

Minix 3.0 isn't really production-ready, but it could make a stable base for a number of projects, so we'd encourage you to try it, whether on a BeagleBoard or a virtual machine, and keep an eye on developments.

Unfortunately, for those wanting to tinker outside of ARM boards on an old PC, in a desktop fashion, there's a major stumbling block: the X Window System, which was working in the previous release (based on a monolithic XFree86 server), has spectacularly broken. At time of writing, the current release of Xorg was being ported, but still wasn't ready, so it's command-line-only for now. You can, however, download previous releases if you really want to try the desktop. Not all NetBSD packages will run, but compatibility is improving and building from source may enable you to get favourite packages onto your system that aren't in the repos.

Inside the (virtual) box

Minix 3 will install on your x86 PC, but hardware compatibility is limited. Any i586 processor or later should be fine, although problems have been reported on Pentium 4, and the system can work with as little as 32MB of RAM. Peripherals are another matter, however: the list of compatible network cards is short, but includes those emulated by *Bochs*, *Qemu*, *VirtualBox*, *Virtual PC* and *VMware*, as well as the BeagleBone's LAN8701A. The remainder of the list is mostly old classics, which will seem familiar if you happen to have experimented with alternative OSes from a decade or more ago: 3Com 509, NE2000, Realtek RTL8139. At least the Intel PRO option is one you're more likely to have on hardware that hasn't been consigned to the attic.

It's probably better to grab a recent *Qemu* or *VirtualBox* from your distro's repository; we did most of our installs on

Ancient history: Tanenbaum vs Torvalds

The Minix 1 source code was available on floppy disks, and in the appendix to Andrew S. Tanenbaum's 1987 book *Operating Systems: Design and Implementation*. Soon a Usenet group grew around Minix with 40,000 subscribers, including Linus Torvalds who added new features. However, he grew frustrated with Tanenbaum's unwillingness to let Minix grow away from its tight educational focus, so in 1991, Torvalds announced on [comp.os.minix](#): "I'm doing a (free) operating system (just a hobby,

won't be big and professional like GNU." When Torvalds was later accused in a book of stealing Minix code, Tanenbaum defended him, but rather waspishly said: "Linus didn't sit down in a vacuum and suddenly type in the Linux source code. He had my book, was running Minix, and undoubtedly knew the history (since it's in my book). But the code was his. The proof of this is that he messed the design up."

The Tanenbaum–Torvalds debate, revisited in the appendix to the 1999 book *Open Sources*:

Voices from the Open Source Revolution was a [comp.os.minix](#) thread over two weeks early in 1992, over the merits of monolithic kernels and microkernels – starting with a Tanenbaum contribution best summed up as 'monolithic kernels are obsolete.' It trailed off inconclusively, but after a few years many claimed 'Linus won,' simply because of Linux's market share. The debate continues, but you can find Tanenbaum's more recent thoughts on the matter at <http://bit.ly/ReliableOS>.

the latter. Click the big New button at the top left of the *VirtualBox* window, and name your VM. We used **MINIX3**. Choose Other in both dropdown menus for OS type and version; on the next screen give it as much RAM as you can spare. We'd suggest that 256MB is adequate for Minix 3, but naturally more is better while you're exploring a new environment and pushing to see what it can do. We kept the default 2GB disk size on one VM install, and expanded to 8GB for another. Be inclined towards the latter if you're going to install *everything* Minix 3 has to offer, but bear in mind 3.3.0's lack of some software from previous releases.

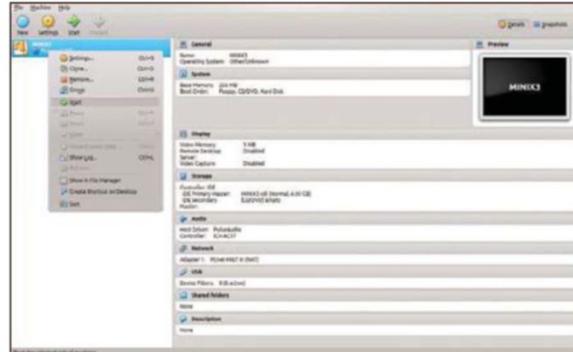
Click Create and you'll see a MINIX3 VM listed in the left pane. Before starting, go to Settings, and tick the Hardware clock in UTC time checkbox, and check that Storage points to your downloaded ISO file. You can now start up the VM from the GUI. On a Core 2 Duo machine, without VT-x/AMD-V/nested paging, we had to start *VirtualBox* with:

```
VBoxSDL --startvm MINIX3 --norawr0 --norawr3
```

Now follow the instructions in the walkthrough (see p77). Once the installation is complete, use **poweroff** instead of **shutdown** to halt the machine. Go back to Settings > Storage in *VirtualBox*, get rid of the ISO, and point at the newly created virtual disk image instead. Now boot with *VirtualBox*'s Start button (or use the workaround on PCs without VT-x and AMD-V virtualisation extensions).

Running on a BeagleBoard

With 3.3.0, the ARM port of Minix is finally integrated in the master code branch, along with the official x86 port. Targeted at the BeagleBoard, with its Cortex-A8-based system on a chip (SoC), it runs with varying degrees of success on the



» **VirtualBox is your best bet for trying out Minix 3, sweeping away hardware compatibility issues.**

BeagleBoard-xM (and its *Qemu*-based emulator), the BeagleBone, and the BeagleBone Black. Pre-built 3.3.0 images are available, but you'll miss out on the latest developments. To build a disk image for your BeagleBoard, make a directory at **~/minix** and **cd** to it – or to wherever you prefer to do the cross-compile – then get the source code:

```
$ git clone git://git.minix3.org/minix minixsrc
```

Now **cd** to **minixsrc/** and make a **.settings** file for BeagleBoard-xM use:

```
# beagleboard-xm
U_BOOT_BIN_DIR=build/omap3_beagle/
CONSOLE=tty02
```

And for the BeagleBone:

```
#beaglebone (and black)
U_BOOT_BIN_DIR=build/am335x_evm/
CONSOLE=tty00
```

Provided you have g++, the GNU C++ compiler, installed, the build tools in Minix will provide you with everything almost else. On Ubuntu you'll also need to:

```
apt-get install zlib1g zlib1g-dev
```

The build tools are based on NetBSD's **build.sh**. Call the ARM-specific version with:

```
./releasetools/arm_sdimage.sh
```

and you should find a **minix_arm_sd.img** waiting for you to copy across to your SD card, for booting your BeagleBoard or BeagleBone:

```
sudo dd if=minix_arm_sd.img of=/dev/mmcblk0 bs=1M
oflag=direct
```

Put the card in your BeagleBoard, boot up and you can log in as root. Ethernet isn't working on the BeagleBone, but on the BeagleBoards run **netconf** and select the LAN8710A. USB support on the BeagleBone is tagged as experimental, and there are known issues with hot-plugging on USB hubs in 3.3.0. Audio and the analogue-to-digital aren't yet working either, but GPIO is.

This is a promising first release of the port, and if you've got a BeagleBoard, it's a good sandbox for trying out Minix, particularly as this port will receive a lot of developer time, so you'll see improvements if you keep checking out the code. You can also try the ARM port under emulation with *Linaro Qemu*, the custom version of *Qemu* for the BeagleBoard-xM. You can find instructions at: <http://bit.ly/LinaroQemu>.

Reliability through architecture

Although at an early stage, the ARM port makes sense for an OS that sells itself on reliability and low resource use. The embedded ARM space covers thousands of different devices



So often *nix problems come down to either permissions or DNS. If you can't get pkgn to work after you install on *VirtualBox* with NAT, put our old friend **8.8.8.8** in */etc/resolv.conf*, instead of **127.0.1**.

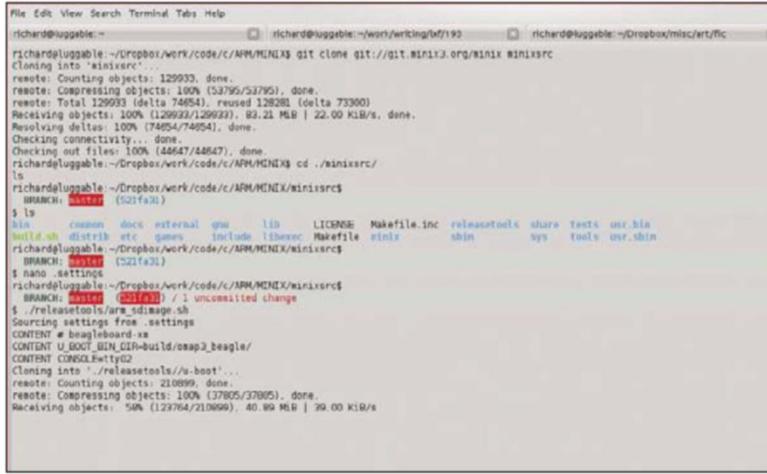
» **If you missed last issue** Call 0844 848 2852 or +44 1604 251045

» where the job of the operating system is, essentially, not to fail over. In consumer devices the mean-time-to-failure should be longer than the life of the device, and Andrew Tanenbaum's stated aim of being finished when "computers don't need a reset button" seems even more applicable here.

Minix's reliability comes from modularity and fighting bloat. There are 6,000 lines of code in the kernel (see *A minimalist codebase*, foot of page). Outside of those 6 KLoC (which manage little more than interrupts, scheduling, and message passing) Minix runs everything else in the OS (all the drivers, from the console to the disk drive) in User Mode protecting against bugs and malicious attacks alike. To access memory, drivers and servers have to ask the kernel. No process is permitted to access anything beyond what it needs: the audio driver cannot access the disk drive, for example. The principle of least authority only allows kernel calls relevant for that class of driver, and allocates time slices to prevent infinite loops bringing down the system.

On top of that driver layer sits virtual memory management, the process manager, file systems and the remainder that traditional Unix regards as part of the kernel. These run in User Mode, with the MMU turned on.

User programs sit above this, but all three layers in User Mode are just – as the kernel is concerned – user processes. A crash in the code would bring down the system under Linux, but Minix 3 would merely need to restart the process. The data store server saves the state from crashed drivers, which is retrieved by the new one starting as a replacement.



```
File Edit View Search Terminal Tabs Help
richard@uggable: ~ richard@uggable: ~/work/writing/bf/193 richard@uggable: ~/Dropbox/minix3/art/mic
richard@uggable: ~/Dropbox/work/code/c/ARM/MINIX git clone git://git.minix3.org/minix3rc
remote: Counting objects: 129933, done.
remote: Compressing objects: 100% (53795/53795), done.
remote: Total 129933 (delta 74854), reused 128281 (delta 73300)
Receiving objects: 100% (129933/129933), 93.21 MB | 22.00 KiB/s, done.
Resolving deltas: 100% (74854/74854), done.
Checking connectivity... done.
Checking out files: 100% (44647/44647), done.
richard@uggable: ~/Dropbox/work/code/c/ARM/MINIX# cd ./minixsrc
richard@uggable: ~/Dropbox/work/code/c/ARM/MINIX/minixsrc$ BRANCH=master
richard@uggable: ~/Dropbox/work/code/c/ARM/MINIX/minixsrc$ nano settings
richard@uggable: ~/Dropbox/work/code/c/ARM/MINIX/minixsrc$ BRANCH=master
richard@uggable: ~/Dropbox/work/code/c/ARM/MINIX/minixsrc$ ./releaseutils/arm_sdimage.sh
Sourcing settings from .settings
CONTENT=beagleboard-xm
CONTENT_U=0.0.0.0
CROSS_COMPILE=arm-build/arm-build/cross_2_beagle/
CONTENT_CONSOLE=tty0
Cloning into '/.releaseutils/u-boot'...
remote: Counting objects: 210899, done.
remote: Compressing objects: 100% (37805/37805), done.
Receiving objects: 50% (123764/210899), 40.89 MiB | 99.00 KiB/s
```

» Building the SD card image for a BeagleBoard uses the NetBSD `build.sh` script and downloads most of the build tools needed on the fly.

Monitoring all of the servers, and restarting them after a crash, if appropriate, is the reincarnation server.

This is safe to do as most failures are caused by random timing errors and races. Tanenbaum's team have tested this, injecting millions of faults to overwrite 100 machine instructions in the running Ethernet driver binary. They injected 800,000 faults into each of the three different Ethernet drivers, causing 18,000 driver crashes – every time, the reincarnation server automatically replaced the driver.

Of course, you could start attacking the kernel yourself – a comprehensive test suite is included – but first get your system the way you like it. As noted in the walkthrough (see p77), `repositories.conf` needs editing to have the correct Minix FTP address. Uncomment the NetBSD repo while you're there. If you've never been on a server where `vi` is the only editing option, I'd recommend installing `bsdgames` on your laptop, and playing command-line Robots, until HJKL keystrokes are in muscle memory. Now run:

`pkgin update`

If there's a problem, particularly via NAT on *VirtualBox*, our best guess is DNS, so put:

`nameserver 8.8.8.8`

in `/etc/resolv.conf`. Now you can use `pkgin` search to find packages, for example:

`pkgin install vim`

to install. If you made a larger hard disk partition, and want to install everything at once, use:

`pkgin_all`

If you want to delve deeper, start with the wiki at <http://minix3.org>, which has a good tutorial on writing device drivers, for example, and plenty of other developer documentation. The user docs aren't too bad, but occasionally pages lag behind by a release version for a while. Naturally, help on updating the wiki would be appreciated by the project as much as more coders – it's actually at the top of the wish list on the wiki!

A bright future for microkernels?

That concludes this overview of Minix 3 – and of microkernels in general. We might be a long way from a usable Hurd, but microkernels have proved their worth in QnX-powered cars [see *Driven by Linux*, p40, LXF187], Cisco routers and elsewhere. Just as a stable alternative for embedded ARM development, Minix 3 is well worth considering. Given its (nearly integrated) NetBSD userland, Minix makes the most usable microkernel OS for those familiar with GNU/Linux, and we confidently expect good things in the next release. **LXF**

A minimalist codebase

A few years ago, Linux passed 15 million lines of code (LoC). Minix 3 has 6,000 (6 KLoC) in the kernel, and about the same again for the drivers (that's traditional kernel code running in User Mode). With a thousand times more code in Linux, we could assume that it has a thousand times more bugs. But it's worse than that.

All code has bugs, typically five to ten bugs per KLoC, although FreeBSD comes out somewhat better with three per KLoC. At FOSDEM, Tanenbaum cited a Stanford study

showing that Linux driver code had three to seven times more bugs than the rest of the kernel – simply because there's no fun in looking at messy driver code, rather than the important (and interesting) algorithms in the core kernel. In Linux 70% of the code is drivers. In Windows, 85% of the crashes are caused by drivers (which are mostly third-party). Running those buggy drivers in kernel space means vulnerabilities can bring down the whole system; in protected memory, isolated from other processes, the

reach of vulnerabilities is strictly limited. Take Netfilter, which replaced Linux 2.2's ipchains to control packet direction – offering options for packet filtering, network address translation and port translation. Netfilter sits inside the kernel, whereas Minix Netfilter sits in user space. If malicious executable code targets Netfilter again (such as the 2.6 'packet of death' kernel vulnerability) in Minix the User Mode process would be compromised, but in Linux the whole system would be owned.

» **Never miss another issue** Subscribe to the #1 source for Linux on page 32.

Installing the Minix 3 ISO

```
Welcome to the MINIX 3 installation CD
=====
1. Regular MINIX 3
2. Regular MINIX 3 (with AHCI)
3. Edit menu option
4. Drop to boot prompt

Choose an option; RETURN for default; SPACE to stop countdown.
Option 1 will be chosen in 0 seconds.
13033+6035+55425+158064+397140=0x9bb7c
WARNING: couldn't open cd9660 (//cd9660/cd9660.kmod)
Loading /mod01_ds
Loading /mod02_rs |
```

```
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,  
2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013  
The NetBSD Foundation, Inc. All rights reserved.  
Copyright (c) 1982, 1986, 1989, 1991, 1993  
The Regents of the University of California. All rights reserved.  
  
For post-installation usage tips such as installing binary  
packages, please see:  
http://wiki.minix3.org/UsersGuide/PostInstallation  
  
For more information on how to use MINIX 3, see the wiki:  
http://wiki.minix3.org  
  
We'd like your feedback: http://minix3.org/community/  
  
# setup  
  
Welcome to the MINIX 3 setup script. This script will guide you in setting up  
MINIX on your machine. Please consult the manual for detailed instructions.  
  
Note 1: If the screen blanks, hit CTRL+F3 to select "software scrolling".  
Note 2: If things go wrong then hit CTRL+C to abort and start over.  
Note 3: Default answers, like [y], can simply be chosen by hitting ENTER.  
Note 4: If you see a colon (:), then you should hit ENTER to continue.  
:-)
```

1 Booting

Boot the ISO image from the default menu choice: Regular MINIX 3. Ignore the warning over cd9660 (this is a known problem and the buggy isofs driver has been disabled), the other modules will load just fine (for a VM on older processors, see the workaround in the main text), followed by the usual disk-mounting, and starting of services and daemons, and the familiar login prompt.

```
abat2      japanese      russian-cp1251  uk
dvorak    latin-america  russian-cp866   ukraine-ko18-u
french     norwegian     russian        us-std
german     polish        scandinavian  us-swop
italian    portuguese    spanish

Keyboard type? [us-std] uk

--- Step 2: Selecting full distribution ---

--- Step 3: Create or select a partition for MINIX 3 ---

Now you need to create a MINIX 3 partition on your hard disk.
You can also select one that's already there.

If you have an existing installation, reinstalling will let you
keep your current partitioning and subpartitioning, and overwrite
everything except your $1 subpartition (<home>). If you want to
reinstall, select your existing minix partition.

Unless you are an expert, you are advised to use the automated
step-by-step help in setting up.

Press ENTER for automatic mode, or type 'expert': E_
```

2 Inline documentation

Log in as root (you won't need to supply a password yet). You're now in the live disc's environment, and can start exploring all that Minix has to offer from here – but if you've got *VirtualBox* running a full installation there's a better sandbox in which to discover Minix 3. Type **setup** to start the install program. Read the four notes that are output, and then hit Enter.

```
Block size in kilobytes? [4]

You have selected to reinstall MINIX 3 in the partition /dev/c0d0p0.
The following subpartitions are now being created on /dev/c0d0p0:

  Root subpartition: /dev/c0d0p0s0  128 MB
  /home subpartition: /dev/c0d0p0s1  512 MB
  /usr subpartition: /dev/c0d0p0s2  rest of c0d0p0

Creating /dev/c0d0p0s0 for / ..
Creating /dev/c0d0p0s1 for /home ..
Creating /dev/c0d0p0s2 for /usr ..

--- Step 7: Wait for files to be copied ---

All files will now be copied to your hard disk. This may take a while.

Remaining: 8223 files. {-----}
/mnt/ast/.exec
/mnt/ast/.profile
/mnt/benchmarks
/mnt/benchmarks/unixbench
/mnt/benchmarks/unixbench/Run
/mnt/benchmarks/unixbench/pgms
```

3 Not Ukrainian?

Your first choice is keyboard layout. Unlike on your Linux distro, the relevant option is UK, not GB. If you can't believe that won't give you a Ukrainian keyboard, try Shift+3 at the next prompt, and you'll get the £ sign. Backspace that £, and hit Enter for automatic disk partitioning, unless you really have a specific partition plan; in which case, type **expert**.

4 Disk partitioning

Even if you choose the automatic partitioning process, you still get to make many choices, provided the correct disk region is chosen (which isn't a problem on *VirtualBox* installations). Just hit Enter, and again at the disk size prompts for partitions and block sizes. Minix gives most of the disk to the **/usr** partition, as this is where most files will live.

```
Saving random data...
1+0 records in
1+0 records out

--- Step 8: Select your Ethernet chip ---

MINIX 3 currently supports the following Ethernet cards. PCI cards detected
by MINIX are marked with *. Please choose:

0. No Ethernet card (no networking)
1. 3Com 501 or 3Com 509 based card
2. Realtek 8029 based card (also emulated by Qemu)
3. NE2000, 3com 503 or WB based card (Also emulated by Rocchs)
4. lan8721ea (on BeagleBone Black)
5. Atttanasi/Atheros L2 FastEthernet
6. DEC Tulip Z1140A in VirtualPC
7. Intel PRO/1000 Gigabit
8. Intel PRO/100
9. - AMD LANCIE (also emulated by VMWare and VirtualBox)
10. Realtek B139 based card
11. Realtek B169 based card
12. Virtin network device
13. Different Ethernet card (no networking)

Ethernet card? [9] -
```

```
WARNING: order matters, duplicates will not be added. If two
repositories hold the same package, it will be fetched from
the first one listed in this file.

This file format supports the following macros:
$arch to define the machine hardware platform
$osrelease to define the release version for the operating system

Remote ftp repository

ftp://ftp.netbsd.org/pub/pkgsrc/packages/NetBSD/$arch/5.1/All

Remote http repository

http://mirror-master.dragonflybsd.org/packages/$arch/DragonFly-$osrelease/stab
ic/All

Local repository (must contain a pkg_summary.gz or bz2)

file:///usr/pkgsrc/packages/All
ftp://ftp[minix].org/pub/minix/packages/3.3.0/i386/All
[q]

pkgin update
Database needs to be updated.
proceed ? [Y/n] _
```

5 Network cards

Your network card – if it's compatible – should be auto-detected with all the VM options well covered, and that includes some older (but well documented) cards. Take a good look at the list if you're planning on a hardware install of Minix 3 somewhere down the line. Now hit Enter to accept the discovered card (which will be AMD LANCE on *VirtualBox* and *VMWare*).

6 Up and running

After power cycling and booting, log in and use **passwd** to set a root password. In **/usr/pkg/etc/pkgin/repositories.conf**, change:
http://www.minix3.org/pub/pkgsrc/packages/\$osrelease/\$arch/All
which should be the only uncommented line, to:
ftp://ftp.minix3.org/pub/minix/packages/3.3.0/i386/All/
and, finally, run **pkgin update**.

Grub: Custom boot menus

Neil Bothwick explains how to change the rather boring default boot menu into something that looks and does the things that you want it to do.



Our expert

Neil Bothwick has a computer in every room, but won't disclose the location of his central server for security reasons.

Bootloaders, such as *Grub*, are software packages that we give little thought to until things go wrong. You turn on your computer, the boot menu (possibly) appears and then your operating system loads. However, there is plenty you can do with the boot menu to tune the way your computer boots and the options available to you.

There are two major versions of *Grub* still in use. The current version is *Grub 2* while the original is often referred to as *Grub Legacy*, because it never reached the version 1.0 milestone; the latest you will find is 0.97. *Grub Legacy* used a configuration file at **/boot/grub/menu.lst** that you were required to edit to add or remove options. That was fine for those of us happy to hack at text configuration files, but not so useful for generating boot menus automatically, so *Grub 2* uses a different system. It still has a configuration file in **/boot/grub**, now called **grub.cfg**, but this file can be created automatically – and usually is – based on what is installed on your computer. We will look at how you can influence this automatic configuration process to tweak the boot menu to suit your needs, whether that be in terms of adding options, speeding things up or adding a more attractive menu for less geeky users.

How the menu is built

The main configuration file lives in **/boot/grub**, but this is usually generated automatically by **grub-mkconfig** (Ubuntu users have **update-grub**, which is a script that calls **grub-mkconfig**). When you run **grub-mkconfig** it looks in two places to determine what it will do: **/etc/default/grub** and **/etc/grub.d**. The former is a file containing some environment variables that *Grub* uses, while **/etc/grub.d** contains a number of shell scripts that are run to generate



► All of the settings you need, and then some, are documented in *Grub's* comprehensive info pages.

the configuration file. The most important scripts are **00_header** and **10_linux**. The first sets up the *Grub* configuration, creating the first part of **grub.cfg** with global settings. Then **10_linux** scans your hard drive for Linux systems and creates menu entries for each kernel in each installed distro. If you dual boot with Windows, **30_os-prober** then adds menu entries for non-Linux operating systems. These scripts are provided by *Grub* and there's no need for you to change them, but you can change the way in which they operate by setting or altering variables in **/etc/default/grub**.

Changing the defaults

The first item you may wish to change is **GRUB_TIMEOUT**, which sets how many seconds the menu is displayed for before booting the default option. If you almost always take the default option, try reducing this to 2 or 3 to speed up the process. Setting it to 0 makes *Grub* boot the default option immediately, but that means the options for recovery or other kernels will not be available to you. Setting it to -1 disables the timeout altogether. **GRUB_HIDDEN_TIMEOUT** performs a similar function when the menu is hidden, as some distros do. This sets how long the boot will pause, waiting for you to press Esc to show the menu. Comment this out if you always want to see the menu. If you don't use Windows, you can save a little time when generating the menu by setting:

GRUB_DISABLE_OS_PROBER=true

Each menu entry will be passed the kernel location, the root filesystem and the location of a matching initrd if one is

Adding themes

We've only looked at altering the content of the boot menu, but *Grub* also lets you change its appearance. You can set a graphical background by setting **GRUB_BACKGROUND** to the full path to a PNG or JPEG file, or you can theme everything on the display.

Themes are contained in directories in **/boot/grub/themes** and you select one by setting **GRUB_THEME** to the name of a theme directory (not the full

path) in **/etc/default/grub**.

The default theme is Starfield, although this is packaged separately in some distros so you may need to install it. Themes are installed to **/usr/share/grub/themes**. Copy any you want to use to **/boot/grub/themes**, set **GRUB_THEME** accordingly and reboot. If you do want to experiment, start with an existing one and modify it. The *Grub* info page documents the components.

found. You can add extra options to the kernel command line by adding them to **GRUB_CMDLINE_LINUX** or **GRUB_CMDLINE_LINUX_DEFAULT**. The first affects all menu entries while the second is not applied to recovery options. Some people dislike the use of UUIDs in **/etc/fstab** and boot menus, as they are hardly human-readable. You can turn them off with:

GRUB_DISABLE_LINUX_UUID=true

There are other options available here, not all documented in the file's comments. Read the configuration section of the *Grub* info pages for a full list.

Custom menu entries

Altering settings in **/etc/default/grub** changes the appearance or behaviour of the standard entries in the menu, but what if you want to add your own? The simplest way to add an extra entry is by editing **/etc/grub.d/40_custom**. This is a script, just like the others used to build the menu, but one that currently adds nothing. Add the menu entries you want to this file to have them added to the end of the menu; do not remove the existing content. For example, you could keep a copy of the Rescatux ISO file in **/boot** and add a menu entry for it by putting this in **40_custom**:

```
menuentry "Rescatux 0.32 - 64 bit" {
    isofile=rescatux-0.32b2.iso
    loopback loop $isofile
    linux (loop)/live/vmlinuz1 findiso=$isofile boot=live config
    quiet splash
    initrd (loop)/live/initrd1.img
}
```

Since the ISO file is referenced twice here, we use a variable to make updating easier. Now when you run **grub-mkconfig**, you will have an entry for Rescatux.

The scripts in **grub.d** are executed in order, so if you want your custom entries added before the Windows options from **30_os-prober**, simply rename **40_custom** to **15_mystuff**. There's also no limit on the number of scripts you can add: just make sure you include the lines from the top of **40_custom** and make your script executable. You can run each script in **/etc/grub.d** in a terminal to see exactly what it adds to the menu.

Writing your own menu entries by hand adds the possibility to making a syntax error, so always check that the menu file is correct before booting it. If you don't and you have a problem booting from it, you will have to press E at the *Grub* menu to make changes. Fortunately for us clumsy typists, *Grub* includes a tool to check your menu:

grub2-script-check /boot/grub/grub.cfg

Dropping a menu entry or two into **40_custom** is easy, but you need to remember to edit it whenever you update whatever it's running. As the scripts in **/etc/grub.d** are simply shell scripts that spit out content for **grub.cfg**, it is possible to make this more sophisticated. For example:

```
for ISO in /boot/rescatux*.iso; do
    echo menuentry "\"Rescatux 64 bit from ${basename
$ISO}\" {""
    echo -e "tisofile=$ISO"
    echo -e "tloopback loop \$isofile"
    echo -e "tlinux (loop)/live/vmlinuz1 findiso=\$isofile
boot=live config quiet splash"
    echo -e "tinitrd (loop)/live/initrd1.img"
```

Grub or Grub 2?

There is still some inconsistency between distros regarding the names of commands and where the files are stored. It was common to add a **2** to the names of *Grub 2* commands, so that both versions of *Grub* could be installed.

For example, **grub-mkconfig** became **grub2-mkconfig**. Some distros still do this so check in **/usr/bin** to see which naming convention is used. Similarly, **grub.cfg** could be in **/boot/grub** or **/boot/grub2**.

echo "}"

done

Save this in **/etc/grub.d**, make it executable and run it. It should give the same menu as the static version, but when you add or replace the ISO with a newer version, it will adapt the menu accordingly. Of course, you could completely replace the existing scripts with your own if you want a truly custom setup. Don't be tempted to edit the existing scripts, as your work would be overwritten by a package update. Instead, clear the executable bit on the scripts you don't want to run and give yours a unique name.

Using your new menu

Changes in **/etc/default/grub** and **/etc/grub.d** are not applied to the *Grub* menu until you run **grub-mkconfig**, which uses everything we have looked at to generate the configuration file. It's best to test your menu syntax first with:

grub-mkconfig | grub-script-check

If this produces no errors, it is safe to write the configuration to the actual menu file:

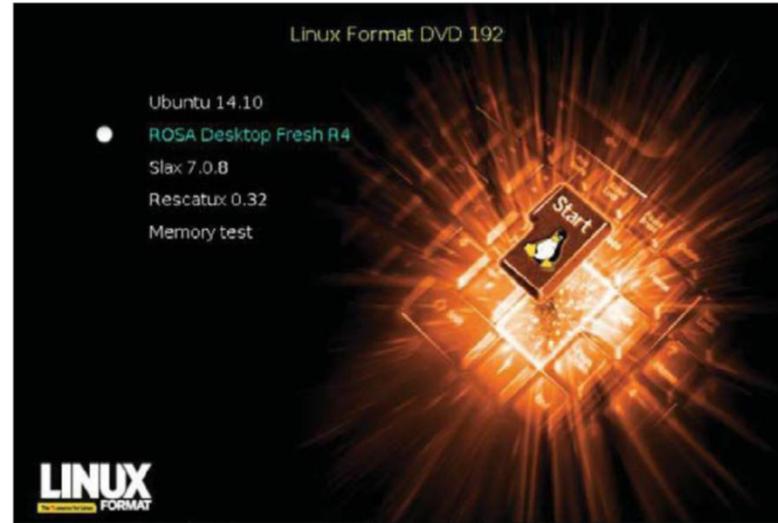
grub2-mkconfig -o /boot/grub/grub.cfg

This is the normal location for the menu file with BIOS systems. With UEFI you may find that it's elsewhere, in which case, **locate** will help you find it.

When you boot your computer, you should now see your custom menu. *Grub 2* certainly introduced some new ways of working, and a lot more flexibility. Because of this, it's possible to make your computer truly unique right from the moment you turn it on. **LXF**



Changing the boot options requires superuser permissions: you're writing to system directories in **/etc** and **/boot**. The commands need to be run as root or prefixed with **sudo**.



► You can completely change the appearance of the boot menu with a theme. You may have seen this one before.

» **If you missed last issue** See www.myfavouritemagazines.co.uk

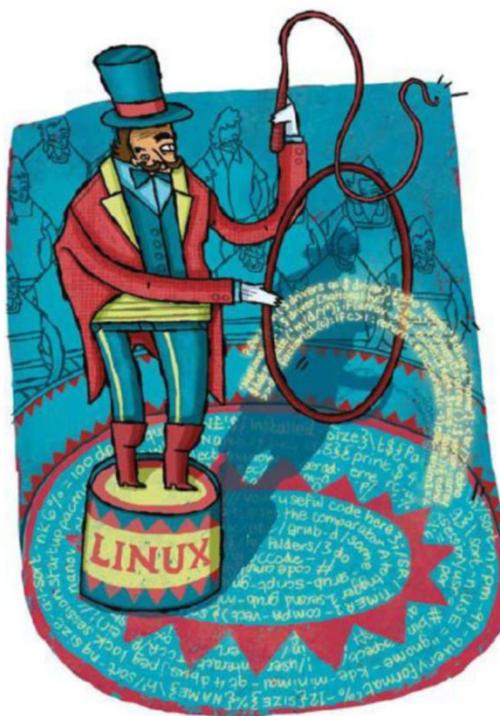
Awk: Advanced text processing

Andrew Mallett explains how to process data from server log files using the sed stream editor for simpler tasks, and the Awk language for complex ones.



Our expert

Andrew Mallett is a Linux trainer with over 700 videos on YouTube (<http://bit.ly/UrbPeng>). You'll also find his courses on www.pluralsight.com.



Assuming that you've been reading recent issues of *Linux Format*, you should be familiar with Awk, since Neil Bothwick has already provided a great introduction to the language [see Tutorials, p74, LXF191]. In this article, we will explore how practical it can be for processing server log data and configuration files.

An introduction to text processing

Before we do that, let's demonstrate the power of text processing with a quick example using the utility tool **grep**. You probably already know that you can show defined shell functions using the command **declare -f**. When you use this command, the output will list the complete function definition, including the name. We have the option of using **declare -F** to list just the function name, but annoyingly, the output includes **declare -f** preceding each function name. The **grep** command can filter the output for us. In order to do this, we simply use:

```
declare -f | grep ^[a-zA-Z]
```

We take the standard output from **declare -f** and filter it with **grep**. The regular expression we use states that we

should only display lines that begin with (this is specified by the carat symbol) a lowercase character or an underscore, (which is specified within the square brackets). Code within a function will be tabbed in and, as such, will not start with a letter or underscore. Only lines that include the function name match the filter so the output is exactly as desired – a list of function names.

Using sed to power your Dockerfiles

Let's move on to a more complex example. Rather than diving straight in to Awk, we'll start by using the **sed** (Stream EDitor) utility to process Dockerfiles.

As Jolyon Brown explained in a previous issue [see Tutorials, p80, LXF191], a Dockerfile can be used to build Docker images. It could start with a base image of Ubuntu and add the SSH Server, for example, or start with a CentOS base image and install Apache. However, in both of these cases we will need to edit the configurations of the given service. First, let's take a look at a Dockerfile that could be used to create an SSH Server Application container:

```
FROM ubuntu
RUN apt-get update && apt-get install -y openssh-server
RUN mkdir /var/run/sshd
RUN echo 'root:Password1' | chpasswd
RUN sed -i 's/PermitRootLogin without-password/' \
    'PermitRootLogin yes/' /etc/ssh/sshd_config
RUN sed -i 's@session\s*required\s*pam_loginuid.so@' \
    'session optional pam_loginuid.so@g' /etc/pam.d/sshd
EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]
```

During the build process of our new custom image we include two **RUN** lines that execute **sed** code. Both use the **sed** command to substitute one text string with another, but the formatting is slightly different.

The first case uses traditional forward slashes to delineate the first string, which it replaces with the second string.

The basic syntax that we use to substitute text in a file is:

```
sed -i 'S/String/Replace/' /etc/ssh/sshd_config
```

Using the option **-i** allows for the file to be edited rather than sent to **STDOUT**. As you can see, we search through the file **/etc/ssh/sshd_config** and substitute the line that contains **PermitRootLogin without-password** with **PermitRootLogin yes**. The original setting is used, by default, to disable password based login for the root user. In such a case root may only log in using public-key authentication. We want to be able to log in as root using the password that

Quick tip

To determine the version of **sed** you are using you can run the command: **sed --version**.

Similarly with Awk we would use: **awk -W version**.

we set earlier on within the Dockerfile. Using `sed` in this manner it's easy to make the needed change in configuration.

In the second example, from the same Dockerfile, we see that we can use delineators other than the forward slash. In this case we use the `@` symbol. We've chosen this because the use of the back slash within the regular expression pattern that makes up the first string makes the information easier to read if we use an alternative delineator. The basic syntax now becomes:

```
sed -i 's@String@Replace@' /etc/pam.d/sshd
```

When we take a look at the working example from the Dockerfile, we use a regular expression as the string to be replaced:

```
session\s*required\s*pam_loginuid.so
```

The `\s` matches any whitespace character and we use the `* quantifier` to indicate that any number of whitespace characters can be used. This takes care of instances where there are, say, two spaces between each word, or other spacing characters, such as tabs. The regular expression will match either, making the number or type of whitespaces unimportant. The replacement string is more easily read as it's a standard string with standard spacing. The purpose of the change to the PAM file here is to ensure that we can still successfully connect even if auditing is required. This is a minimal configuration and all the required elements may not be present; setting the module to optional means we do not consider the success or failure of the PAM module.

We can see that using `sed` in this instance has provided a relatively simple mechanism to edit configuration files during the build process of Docker images. Where changes are quite minimal this is preferable to uploading completely new configurations during the build process.

Similarly, we can delete lines from files as well as substituting the line contents; it's just a matter of using the command `d` (for delete) instead of `s` (for substitute). However, using the delete command requires that we specify the range of lines to work with, whereas previously we worked with the complete file, line by line. The range is specified before the `d` through the use the forward slash at the start and the end of the range. These delimiters *must* implement the forward slash, unlike the string delimiters we used previously with the substitute command.

In the following example, we create the Docker image from the CentOS 6 base installation, install the *Apache* HTTPD server and remove an unneeded module from the web server configuration:

```
FROM centos:centos6
RUN yum install -y httpd
RUN sed -i '/LoadModule\s*userdir_module/d' /etc/httpd/conf/httpd.conf
RUN echo "Welcome to My Site" > /var/www/html/index.html
EXPOSE 80
ENTRYPOINT ["/usr/sbin/httpd", "-DFOREGROUND"]
```

Of course, not all of you will be using Docker – at least, not at the moment. (I am sure that given enough time, we can convince you of the benefits.) However, `sed` can be used in many other contexts.

One way that I often use `sed` is to supplement `-i`, for the in-place edit, with an extension enabling a backup prior to that edit. Many configuration files in Linux are splattered with comments and extra blank lines. Although I am not opposed

```
#manycastserver 239.255.254.254      # manycast server
#manycastclient 239.255.254.254 autokey # manycast client

# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8
```

► The standard CentOS `ntp.conf` includes many blank and commented lines, making it difficult seeing the wood for the trees. We can fix this using `sed`.

to comments, this can make understanding the configuration a lot more difficult – and, in some cases, can encourage you to duplicate a setting as it isn't easy to see where it was previously set.

A simple illustration of this point is the file `/etc/ntp.conf`. This is the time server configuration, and has 53 lines on my CentOS 6 box; however, only 11 lines actually do anything. While this is not a particularly extreme case, it highlights the problem. I would always create a backup of the file first, which then becomes my commented file while the cleaned original file becomes the working configuration:

```
sed -i commented '/^#/d;^$d' /etc/ntp.conf
```

Here, `sed` uses two expressions, separated using the semicolon (`:`). The first expression deletes lines that start with a `#` – that is, commented lines. The second expression deletes blank lines or, as is represented by the regular expression `^$`, lines that begin with an end of line marker. When this is run as our root user we will reduce the contents of the `ntp.conf` to 11 lines and keep the original file. The original file with all the comments and extra lines intact is now called `/etc/ntp.conf.commentated`.

Note the use of the extension that immediately follows the `-i` option. There can be no extra white space between the option and the file extension you wish to add.



When you use `sed` -i to edit a file it may be prudent to omit the -i initially to ensure that the output that you get matches the output you expected.

Awesome Awk

If `sed` is `grep`'s big brother, you could say that Awk is the daddy of them both. In his earlier article [Tutorials, p74, LXF191], Neil provided an introduction to Awk and its capabilities. Here, we'll put those capabilities to use. First, we will see how we can use Awk to enhance the output of the `lastlog` command before moving on to processing XML and then large text files to summarise logs.

To start with, we will need to make sure that we are familiar with `lastlog`. If we use `lastlog` without arguments, it will display the last login time of all accounts, including service accounts that have never logged in. The output is a little cluttered, to say the least. Or we can use the command with options to display the last login time for just one user: `lastlog -u bob`, for example. Alternatively, we could display only user accounts that have not logged in within the last 90 days: `lastlog -b 90`.

This is great but it still displays accounts that have never logged in. Ideally, we would like a report that printed just the

» If you missed last issue Call 0844 848 2852 or +44 1604 251045

- » account name and last login date, as well as excluding those accounts marked as having never logged in.

Initially, we will simply use Awk to filter accounts that have never logged in. This achieves little more than we could do with *grep*, but it illustrates the way in which Awk can be used to inverse the search:

```
lastlog | awk '!Never/ { print }'
```

We send the output from **lastlog** directly to Awk. The Awk statement starts with a range. We specify that range to be the inverse of rows that contain the string **Never**; in other words, we will exclude rows that include the string **Never**. The Awk body then just prints each row that matches the range, so we will see all accounts that have logged in at least once.

Alternatively, we could extend the range to exclude the root account and to remove the header line **Username**:

```
lastlog | awk '!/(Never/ || /^root/ || ^Username/) { print }'
```

We use the parentheses here to group the two ranges together so they can be negated as one. The two vertical bars (**||**) mean a logical OR. We don't process lines that either contain the string **Never**, or start with **root** or **Username**. Even though these multiple exclusions could be written as a **grep** statement we have already passed into the area where Awk will achieve results more simply.

What we have managed so far is okay for a single line of command-line code, but we are going to need to become a little more adventurous with Awk if we want to create a really desirable result. Let's start by creating an Awk file that reduces the amount of syntax that we have to type on the command line to make reuse of the code easier. This will demonstrate some valuable Awk techniques. The Awk file we shall be working with is as follows:

```
BEGIN {
printf "%8s %11s\n","Username","Login date"
print "====="
}

!/Never logged in/ || /^Username/ || /^root/ {
cnt++
if (NF == 8)
printf "%8s %2s %3s %4s\n", $1,$5,$4,$8

else
printf "%8s %2s %3s %4s\n", $1,$6,$5,$9
}
END {
print "====="
print "Total Number of Users Processed : ", cnt
}
```

To run this, we just need to be in the same folder as the Awk file. Here are two examples of the ways in which we could use it:

```
lastlog | awk -f lastlog.awk
lastlog -b 60 | awk -f lastlog.awk
```

```
[root@centos6 ~]$
[root@centos6 ~]$
[root@centos6 ~]$ sed -i.commented '/^#/d;/^$/d' /etc/ntp.conf
[root@centos6 ~]$ ls /etc/ntp.conf*
/etc/ntp.conf  /etc/ntp.conf.commented
```

The first example processes all users; the second example processes just those who haven't logged in for the last 60 days. You can start to appreciate the power of Awk and its data processing and formatting abilities when you compare the output of **lastlog -b 60** with this example.

The Awk file itself contains three sections. The first and last are named rather fittingly: **BEGIN** and **END**. The main body section is unnamed. The **BEGIN** and **END** sections run just once, whereas the main body runs for each line in the matched range.

The **BEGIN** section is where we can set variables, such as delimiters, if required; or, as is the case here, heading information. Using **printf** rather than just **print** enables us the format the information as needed.

The **END** section is used to produce a footer and usually has summary information. Here we **print** the number of users processed by looking at the value of the **cnt** variable that's incremented in the body.

Now comes the main body. Here, we are able to see many elements that the Awk language supplies. The main body itself is defined within the braces (the curly brackets). Immediately prior these brackets, we define the range in the way that we discussed earlier. The main body only works on the lines that match the criteria we set in the range.

The first lines of the body defines and increments the variable **cnt**. We use this as our counter to use within the **END** code. On the first iteration the variable will be undefined and as such will effectively have a value of 0, which we increment to set a value of 1. The next matching row will take us to 2, and so on.

We implemented the **If (NF == 8)** statement to ensure we print the correct fields. Logins from remote clients include nine fields and those from local consoles only include eight. The number of fields within a row is held within the **NF** variable. The statements are used to print the required fields depending on whether we have eight or nine fields in a row.

Using Awk to process XML data

Next, we'll look at accessing XML data with Awk. Along the way, we will discover that although the default record that we look at with Awk is a single line, we can adjust the RS variable to make a record more than one line.

In this scenario, we are storing *Apache* web server Virtual Host information within a single configuration file, but we need to be able to print single and complete virtual host records for any given host. Virtual host definition begins with an opening tag similar to **<VirtualHost *:80>** and closes with the ending tag **</VirtualHost>**. For the example to work, we need to ensure that we have a blank line between each new Virtual Host and the previous host ending. If this is not the case, we can use **sed** to insert a new blank line after each **</VirtualHost>**. We will assume that the virtual hosts are all defined in the file **virtualhost.conf** and that blank lines do not

» **Using sed to clean this unwanted clutter is easily done; as is deleting too much – so test without the -i first and the file will be left untouched. Output will only be shown in the console.**

» **Got questions?** Get your answers from Bothwick the Grey on page 92.

```
andrew@pxepi ~ $ lastlog | awk '!/Never/ { print }'
Username      Port      From          Latest
root          tty1
pi            pts/0    192.168.0.57   Tue Nov 26 11:16:46 +0000 2013
andrew        pts/0    192.168.0.58   Sat Nov  1 14:05:06 +0000 2014
```

► It becomes easy to emulate other tools, such as grep, using Awk.

exist after each definition. The following code will edit the file for you, adding the blank lines:

```
sed -i '</VirtualHost>/G' virtualhost.conf
```

The example virtual host file that we will be working with looks like this:

```
<VirtualHost *:80>
```

```
DocumentRoot /www/example
```

```
ServerName www.example.org
```

```
# Other directives here
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
DocumentRoot /www/theurbanpenguin
```

```
ServerName www.theurbanpenguin.com
```

```
# Other directives here
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
DocumentRoot /www/linuxformat
```

```
ServerName www.linuxformat.com
```

```
# Other directives here
```

```
</VirtualHost>
```

Now that we have the correctly formatted file, we can use the following Awk file, **vh.awk**, to enable us to search for named entries:

```
BEGIN { FS = "</VirtualHost>"; RS = "\n\n"; }
```

```
$0 ~ searchString { print }
```

The **BEGIN** block defines the field delimiter as the closing Virtual Host tag. This delimits entries in each record. A record is normally represented by a line, but we change that to be two consecutive new lines. The main block will print records, now defined as the complete Virtual Host definition, by comparing each record (**\$0**) against a variable that we will populate at runtime (**searchstring**). The Awk code to run this would be similar to this:

```
awk -f vh.awk searchString=www.example.org virtualhost.conf
```

Note that we supply the value to the variable at runtime. The corresponding result should look similar to this:

```
<VirtualHost *:80>
```

```
DocumentRoot /www/example
```

```
ServerName www.example.org
```

```
# Other directives here
```

```
</VirtualHost>
```

```
{ ip[$1]++ }
```

```
END {
```

```
for (i in ip)
```

```
print i, " has accessed ", ip[i], " times." }
```

The **BEGIN** block simply prints the header information. The main block creates a new array for each occurrence of field 1, the client IP address. In this way, we have an element in the array (**ip**) named after each client IP address used to access the server. The value of the individual array is incremented each time the field is matched. This time it's the **END** block that does most of the work, utilising a **for** loop to iterate through each named element of the array **ip** and print its value. When you use the command:

```
awk -f count.awk access.log
```

you may expect an extract of the output to look similar to that below. Bearing in mind that the data was from a production server, we have modified the first octet of the client IP address:

```
xxx.157.100.28 has accessed 1 times.
```

```
xxx.180.86.233 has accessed 10 times.
```

```
xxx.241.226.216 has accessed 2 times.
```

```
xxx.99.52.100 has accessed 12 times.
```

It's a simple matter to edit the Awk file to display the HTTP access code, which is field 9 of the log. In this way we can see the amount of web access to the server during the period the log covers. The output from my log showed these results:

```
Log access
```

```
The access code: 200 has occurred 23825 times.
```

```
The access code: 206 has occurred 48 times.
```

```
The access code: 301 has occurred 60 times.
```

```
The access code: 302 has occurred 21 times.
```

```
The access code: 304 has occurred 2273 times.
```

```
The access code: 403 has occurred 133 times.
```

```
The access code: 404 has occurred 4382 times.
```

```
The access code: 501 has occurred 63 times.
```

The 403 errors are forbidden activities where security is needed and failed; error 404s, as you probably all know, are page not found; 2xx codes are success; 3xx codes are normally redirections; and 5xx codes relate to CGI errors. Processing 30,000 lines takes seconds with Awk, showing how easily we can start to assimilate the information. [LXF](#)

Analyse log files with Awk

Finally, let's look at how we can leverage the power of Awk to read through a web server access log and print the number of times each client has access to the web server. The first field in an access log defines the client IP. We can utilise Awk arrays to count the accesses of each client. We will work with a log file that has 30,000 lines: a typical real-life example.

We will need an Awk file again – as we have seen, this is quite normal. This time we call it **count.awk**:

```
BEGIN {
```

```
print "Log access" }
```

```
File Edit View Terminal Tabs Help
dal:~/Desktop # lastlog -b 60 | awk -f lastlog.awk
Username Login date
=====
geeko 2 Jul 2013
bob 6 Jul 2013
=====
Total Number of Users Processed: 2
dal:~/Desktop #
```

► As you discover how useful Awk is to customise the output of commands to meet your needs, you will create a plethora of tools using it.

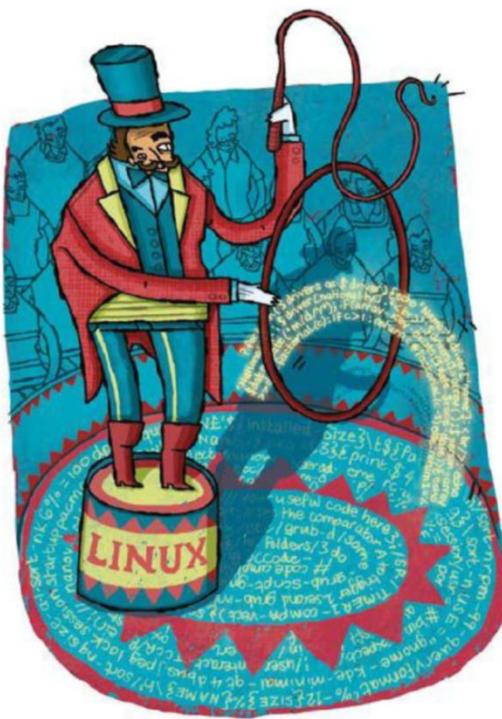
Cython: Speed up Python

Jonni Bidwell explains how to feed some Cython-flavoured accelerant into your system, using image compression as a working example.



Our expert

Jonni Bidwell
is all about getting things done in a timely manner.



Python is a great language. It has a clean and easy-to-learn syntax and you can do an awful lot in a handful of lines. It's just not very fast, which, depending on your purposes, could be a deal-breaker. The main reason for this is that Python is interpreted: it is read line by line and converted on the fly to intermediate bytecode which gets shuffled around and eventually executed on the CPU. This takes time, but it makes life easier: there's no need to compile your code every time you change something, and there's no need to type your variables.

The interpreter will figure out which data type everything should be, and even if you change, say, a list into an integer, it will accommodate your changes without complaint. If you really want your Python code to go fast, then rewrite it in C and fast it will be. This is easier said than done, though: C is hard, and more often than not you'll only be interested in accelerating a handful of bottlenecks in your code.

Enter Cython, commonly misconceived as a Python-to-C translator. On some level this is true: Cython will take your Python code (slightly modified), and spit out a C file which you can compile and then import as an extension module, availing you of turbo-charged versions of all the functions in

your original code. However, you still need your original code: the emitted module is there to convert the relevant parts of it to native machine code, rather than Python bytecode.

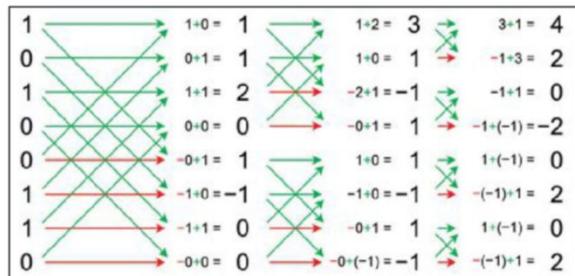
The Cython language is pretty much a superset of Python, so (excepting a few specialised modules and functions) any valid Python is also valid Cython, and as such can be saved as a PYX file and fed to the Cython binary. However, for optimal 'cythonising' one needs to use some of the extra Cython keywords, which can type variables (including function parameters and return types) and provide faster array access.

Many programs won't really gain anything from this Cython treatment, and if you're not careful you can end up actually slowing things down. For example, if your program spends most of its time drawing graphics, or is heavily I/O dependent, these are not things Cython can help you with. However, if your program is spending most of its life looping over arrays, shifting bits back and forth and doing arithmetic, then you are in luck.

Compressing data

We're going to use Cython to speed up a crude implementation of the fast Walsh-Hadamard transform. We are going to use the transform to lossily compress greyscale image data, although the principle applies to any data. In the early days of satellite imagery, NASA used techniques like this since the transform relies only on computationally cheap addition and subtraction operations, and thanks to some mathematical trickery, the number of these operations can be reduced (down to $O(n \log n)$ from $O(n^2)$ if you care about such things).

An 8-bit greyscale image can be represented as a list of unsigned integers from 0 to 255, ie bytes. Each byte corresponds to the intensity of each pixel, and so a 256x256



» **In-place addition and subtraction calculates the Walsh spectrum without having to multiply by a large matrix.**

Quick tip

If you've **cdef'd** everything and still want more speed, you can pass directives (such as the infamous **-O3**) to the compiler. Check the official docs – <http://lbit.ly/CythonDocs>.

Image credit: Wikipedia

image will take up 65,536 bytes, or 64kB. The Walsh functions are a well-known family of functions which take on the values 1 or -1. By summing various component Walsh functions, it's possible to compose any discrete-valued function. For example, a row of pixels in our image, or even the whole image, could be exactly reproduced by, say, summing one Walsh function 300 times, subtracting 84 of another, adding 6 of yet another, then subtracting 2 of yet another other. The Walsh-Hadamard transform will tell you exactly which coefficients go with which functions quickly and efficiently.

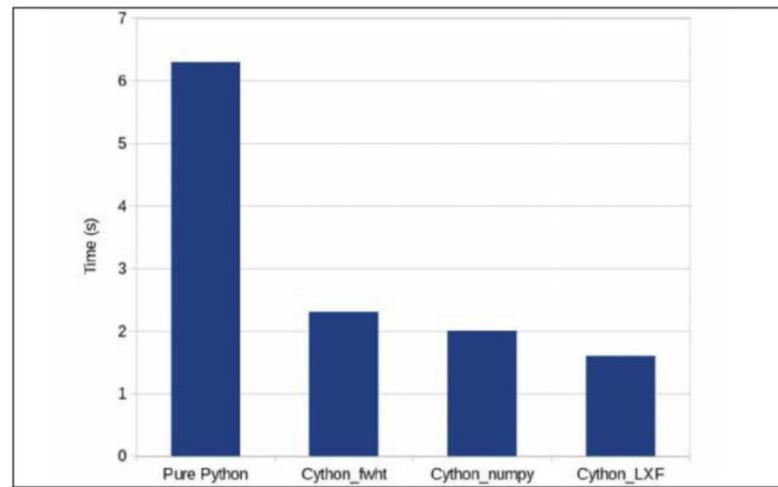
In practice, unless you're working with contrived data, there's no benefit to storing the parent function in this way (you usually need to sum as many functions as you have pixels, or data points). However, if you're not too worried about losing some data, then you can often get a very reasonable approximation of your data by discarding those functions with smaller Walsh coefficients. We won't worry too much about storing or even how to store the approximated image. Instead, we'll make some educated assumptions about its file size – in particular that each coefficient will take 10 bits to store (so that it can take values between -511 and 511) in addition to some bits for each index. We can show what the compressed image looks like, but it will still be represented as an uncompressed array in Python.

The Walsh-Hadamard transform

The Walsh-Hadamard transform is commonly represented as a matrix transform, where a power-of-two-sized square matrix multiplies a power-of-two-sized column vector (our data). The matrix is orthogonal and (when an appropriate scaling factor is used) unitary, so that the transform can be reversed by applying it again. The matrix is an example of a Hadamard matrix, the entries of the matrix (when the scaling factor is excluded) are all +/- 1, and the rows form the Walsh functions. (Fun fact: these were originally discovered 20 years before Walsh was born, in the context of eliminating crosstalk along parallel telegraph wires.)

The fast Walsh-Hadamard transform exploits the recursive structure of the Walsh matrix (it can be defined as a tensor product of 2x2 matrices) to perform the computation much quicker using some neat in-place calculations summed up in the diagram shown on the opposite page.

In the following code, we cheat a little here and use the `log2` function from NumPy. Don't worry too much about the logic arcana surrounding `j` and `k` below. It's just a neat way to recreate the butterfly structure shown in the diagram.



The first optimisations bring the most benefit. After that, it's easy to spend hours trying to save a few milliseconds.

The algorithm works directly on the input, summing and subtracting pairs of entries, and so doesn't need to return anything as a result:

```
import numpy as np

def fwht(arr):
    n = len(arr)
    b = int(np.log2(n))

    for bit in range(b):
        for k in range(n):
            if k & (1 << bit) == 0:
                j = (1 << bit) | k
                tmp = arr[k]
                arr[k] += arr[j]
                arr[j] = tmp - arr[j]
```

The bitshift operators `<<` and `>>` aren't particularly quick in Python, but in C they correspond to a machine level operation and are much quicker than the equivalent literal multiplication or integer division by powers of two.

Our compression algorithm will read, using the Python imaging library, a greyscale image as a 1D array. We will divide this array into chunks and perform the transform on these chunks. We require a function to select and store the largest coefficients resulting from each of them. It makes sense to do some shifting and rounding here too; you can see the result in »

Cythonic decorations

As well as typing variables, we can also specify input or return types for functions. To do this, we define the function with `cdef` and then specify its return type before its name. For example, our core function `fwht` doesn't return anything, and hence should be typed `void`. After we have optimised the stuffing out of `fwht`, it then takes a memory view of C `ints` as input, so it's defined:

```
cdef void fwht(int[: arr)
```

Using `cdef` means that your function won't be available to other Python modules, but you can use `cpdef` (which will incur a slight

overhead) if you need your function to work from outside too. By `cimport`-ing the `cython` module, we can access a few decorators which change behaviours at the function level. For example, to turn off profiling for an individual function, use:

```
@cython.profile(False)
def too_cool_for_timing:
```

You'll find that this is particularly useful when used in conjunction with the `inline` keyword, which is used to 'unroll' small but frequently used functions, and for reducing the overhead

associated with the function call. You will need to put the `inline` keyword right after `cdef`.

Finally, there are a couple of 'dangerous' things that are quite popular, namely:

```
@cython.boundscheck(False)
```

and:

```
@cython.cdivision(True)
```

which respectively deactivate out-of-bounds checking for arrays and checks for division by zero. You really should make sure that your code is correct before doing this, since they have the potential to corrupt memory.

» If you missed last issue Call 0844 848 2852 or +44 1604 251045

- the function **squishChunk()** in the files on the **LXFDVD**. Decompression, via the **expandImage()** function, involves taking each chunk, collating the coefficient indices and magnitudes into a vector, and then performing the transform again, and shifting everything back to the 0-255 range. The chunks are then rejoined and we use the **show()** method to display the resulting lossily compressed image. This method requires you to be running an X server, since it uses the *xv* program (which you also require) to display.

You can test everything works by copying the directory on the **LXFDVD** to a local folder and from there running:

```
$ python proftest.py
```

This program will compress then expand a photo from the Philae module's new home: you can see it on the page opposite. You can experiment with the **chunksize** and **nterms** parameters at the beginning of the **fwht_python** file.

The initial values (32 and 8) give a nominal compression ratio of just over 2:1, although this is meaningless as we aren't storing the compressed data. It could also be vastly improved by varying the number of terms for each chunk – areas of the same colour need only a single term. We can profile this code using the `cProfile` module by running:

```
$ python -m cProfile proftest.py
```

This lists every single function involved in the program, including all the weird functions involved in decoding a PNG image, so we can filter this to show our own efforts by adding **I grep fwht** to the above. On a dusty **LXF** office machine, the whole execution took about five seconds, just over three

Using the `-a` option generates HTML files which show you the clean, white C-like code and the dirty, yellow Python code.

seconds of which was spent in the **fwht** function: pretty reasonable, given that **fwht** is the heart of our program.

We ought to be able to speed this up quite a bit by using NumPy arrays instead of lists. NumPy arrays can be initialised with zeros, but it's slightly quicker to avoid this step (so the array initially contains whatever random data was in the memory allocated to it) if you know the values will be filled in later. You also need to specify a data type for the array, and it's better to not use methods for Python lists such as `len()`. You'll find the code in the file `fwht_numpy.py`. You'll also find that it takes about twice as long to run – benchmarking is full of surprises. Despite this slight disappointment, we'll stick with our arrays – Cython might do a better job with them.

Enter Cython

It makes sense to initially concentrate our efforts on speeding up the **fwht()** function, which is at present quite readable. A simple first step is to specify data types for all the local variables in this function. Although they are all integers, and you can get away with declaring them as such, the **for** loop indices have a special type **Py_ssize_t** so we may as well use it. Add the following lines at the beginning of the **fwht** function in **fwht_numpy.py** and save the file as, say, **fwht_cython1.pyx**:

```
def fwht(arr):
```

```
cdef int n = arr.shape[0]
cdef int b = int(np.log2(n))
cdef Py_ssize_t bit,k
cdef int i,tmp
```

Now run:

```
$ cython -a fwht cython1.pyx[bl]
```

\$ cython -a lwnl_cython1.pyx [b]
This will generate some very messy C in a file called **fwht_cython1.c**. The **-a** switch tells Cython to additionally generate a similarly named HTML file which you should look at. The lines with your newly typed variables are white, whereas most of the rest of the code will be various shades of yellow. You can even click on each line to see how it looks in C, and in so doing you will discover that the yellow lines correspond to lengthier or more involved code. We will still want to do some benchmarking, so add the following decorator at the top of the file:

```
# cython: profile=True
```

Getting your Cython code compiled is a little bit of effort. You can do it manually, but it's easier to use the **`cythonize`** function and the `distutils` module. Create a file **`setup.py`**

Benchmarking

It's easy to take benchmarks too seriously – graphics card enthusiasts have been doing so for years. In our tutorial we use the cProfile module which enables you to count and time each function call. This can provide valuable data about where the bottlenecks in your code are, which might not be immediately obvious.

CProfile is designed to be as lightweight and unobtrusive as possible, but if you have a tiny function that's called millions of times, then that's millions of tallying calls and they all add up. If the function is really small, this means that

more time is spent benchmarking than is spent doing whatever it is the function does, and so the result is largely meaningless. If you're confident a small function can't be sped up any more, it's best to disable the profiler. If you're not so confident, then by all means continue to benchmark such small functions, but rest assured that they will work a lot faster without the profiler interfering.

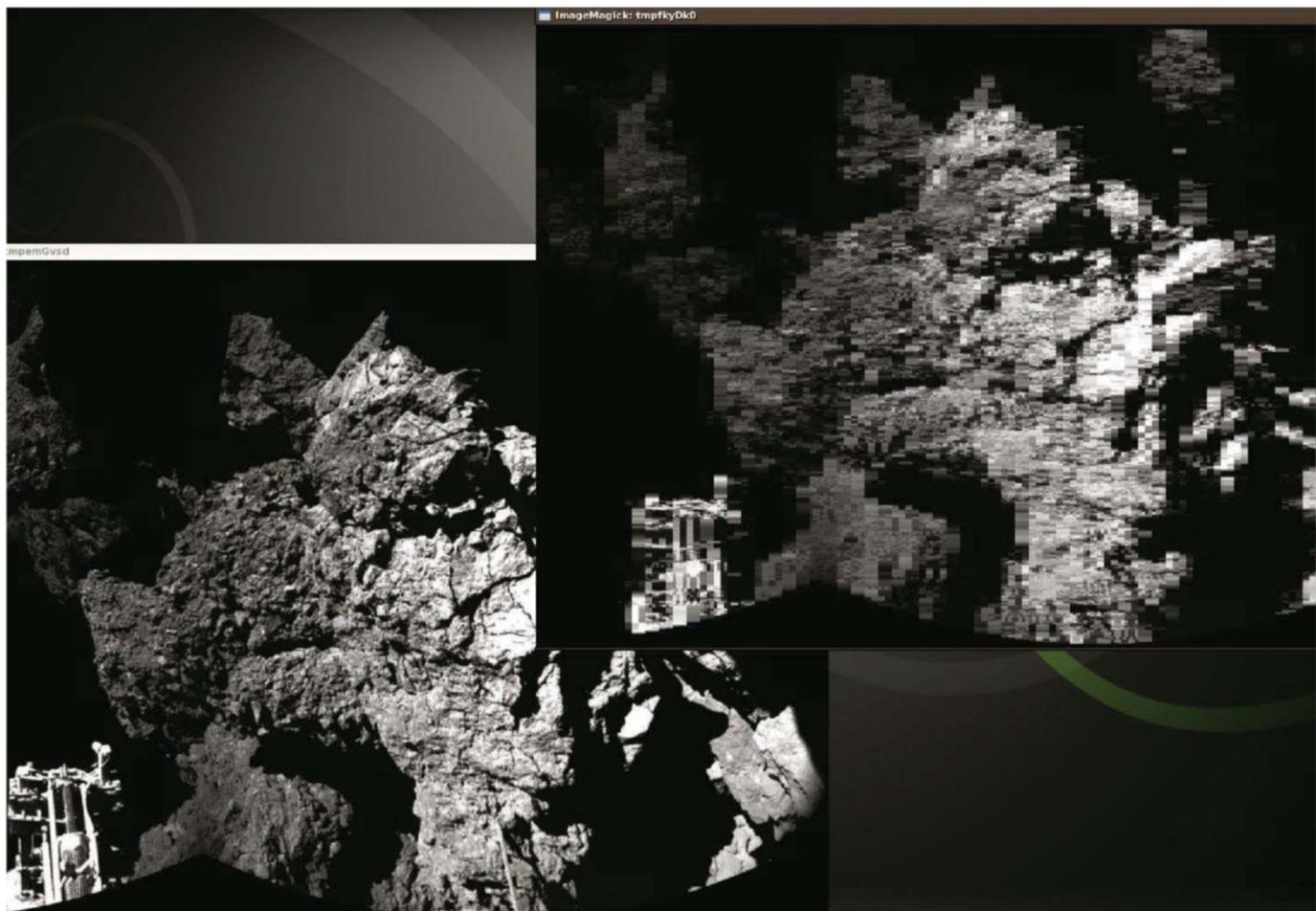
If you just want to measure 'wall time', which is the total time it takes for your code fragment to run, then the **timeit** module may be more

appropriate. For example to test (from the interpreter) a function called **testfunction()** from a module **testmodule** three times:

```
import timeit  
timeit.timeit(stmt='testmodule.testfunction()',  
    setup='import testmodule', number=3)
```

The default for **number** is a million, which is why it's a good idea to specify your own value here. You will need to specify your module in the **setup** parameter even if you have previously imported it, since **timeit** will not inherit this namespace.

» **Feed your inner mathematician** Learn how to program in R on page 88.



(you can find an example on the [LXFDVD](#)) with the following contents:

```
from distutils.core import setup
from Cython.Build import cythonize

setup(
    ext_modules = cythonize("fwht_cython1.pyx")
)
```

Now if you run:

```
$ python setup.py build_ext --inplace
```

everything will be built and you can modify [proftest.py](#) to use your new **fwht_cython1** module. Benchmarking this (still using NumPy arrays) on our machine actually slowed things down – the **fwht()** function alone took about 20 seconds. But don't lose heart: the reason for the slowdown is that C has to access **arr** through Python and NumPy methods.

Memory views

Recently Cython introduced a new way to access array data through something called memory views. These enable C to directly access the array data in memory, and hence are damned fast. But they don't work with Python lists, which is why we are stuck with our slower NumPy arrays. We'll make **fwht()** work on a memory view by changing the definition line:

```
def fwht(int[:] arr):
```

The functions **squishChunk()** and **expandChunk()** will need to be modified too. So let's go ahead and define **f_view** in **squishChunk()** like so:

```
cdef int[:] f_view
f_view =
```

Replace all further references in the function to the array **f** with **f_view**, with the exception of the **enumerate** call, which is a Python function. Likewise define **fbar_view** in **expandChunk()**, and replace all **fbar** references except the **return fbar** statement. While we're at it, we may as well type all the **js** and **ns** and whatnot as **ints** too. Now re-run **setup.py** and benchmark. Now we're cooking with gas – the total execution time was less than three seconds, most of which is now spent in the **squishChunk()** function. The bottleneck here is ranking our coefficients and powers, so let's separate that into a separate **rankArray()** function, which is less reliant on Python constructs:

```
def rankArray(int[:] F):
    cdef int n,j
    n = F.shape[0]
    Franked = np.empty([n,3],dtype=np.int32)
    cdef int[:, :] Franked_view = Franked

    for j in range(n):
        Franked_view[j,0] = j
        Franked_view[j,1] = F[j]
        Franked_view[j,2] = - abs(F[j])
    Franked = Franked[Franked[:,2].argsort()]
    return Franked[:nterms,:2]
```

Now change the **return** line in **squishChunk()** to use this, and check the benchmarks. On our machine this shaved off nearly a second, and we were quite happy about that. From here on in it's diminishing returns, but we've provided as much optimisation as we can in the file **fwht_cython.pyx** on the [LXFDVD](#). This took total execution time down to 1.6 seconds – see if you can do better! [LXF](#)

A postcard from comet 67P in both the original (left) and heavily compressed version (right). Landing stuff on extraterrestrial bodies is pretty much the only way to make space engineers hug each other.



R: Get started

Mihalis Tsoukalos provides an introduction to the statistical programming language, and explores how to use it to analyse system-monitoring data.



Our expert

Mihalis Tsoukalos enjoys statistics and visualising monitoring data. As well as being a mathematician, he's a Unix admin, a programmer and a database administrator.

R is a GNU project based on S, a statistics-specific language and environment developed at the famous Bell Labs. Essentially, you can think of R as the free version of the S language. The R system distribution supports a large number of statistical procedures, including linear and generalised linear models, nonlinear regression models, time series analysis, classical parametric and nonparametric tests, clustering and smoothing.

In order to be as generic as possible, we will only use the command-line version of R in this tutorial, but don't be afraid of the language even if you don't feel comfortable with mathematics, since there are GUIs for it that you can use. The most popular of these (and my personal favourite) is *RStudio* (www.rstudio.com).

Installing R

You can install R on a Debian 7 system by typing:

```
# apt-get install r-base
```

You can then run R and go to its shell by just typing **R** at the Unix shell. The following output shows how to do simple calculations in R:

```
> 4 + 5  
[1] 9  
> 4 * 4  
[1] 16  
> 4 ^ 3  
[1] 64  
> 4 ^ 10  
[1] 1048576  
> 4 / 3  
[1] 1.333333
```

R can automatically read data from structured text files using the **read.table()** command. The single most useful R command for getting a general overview of a data set is the **summary()** command:

```
> data <- read.table("uptime.data", header=TRUE)  
> summary(data)  
 X1min      X5min      X15min  
 Min. :0.00000  Min. :0.01000  Min. :0.05000  
 1st Qu.:0.00000  1st Qu.:0.01000  1st Qu.:0.05000  
 Median :0.00000  Median :0.01000  Median :0.05000  
 Mean  :0.02028  Mean  :0.02491  Mean  :0.05553  
 3rd Qu.:0.00000  3rd Qu.:0.02000  3rd Qu.:0.05000  
 Max. :2.47000  Max. :2.15000  Max. :1.05000
```

You can find more information about the **read.table()** command by typing **help(read.table)**.

Creating new R functions

First, let's look at the R code required to implement two new functions: one for finding Fibonacci numbers and one for finding the factorial of an integer. When defining your own functions, make sure that they have unique names. The code for calculating Fibonacci numbers is as follows:

```
myFibo = function(i)  
{  
  if ( i == 0 )  
  {  
    return(0)  
  }  
  if ( i == 1 )  
  {  
    return(1)  
  }  
  if ( i == 2 )  
  {  
    return(1)  
  }  
  
  return (myFibo(i-1) + myFibo(i-2))  
}
```

The code should look pretty familiar to you. As you can see, you don't need to initialise or declare variables. Be careful though, because sometimes this can cause bugs or other nasty problems.

After saving the code, you can load it using **source()**, provided that your working directory is the directory where **fibonacci.R** is located (otherwise type the full path):

```
> source("fibonacci.R")  
> myFibo(4)  
[1] 3  
> myFibo(15)  
[1] 610  
> myFibo(26)  
[1] 121393
```

If everything is OK, R will print no output after executing the **source()** command. You can also see that R automatically prints the result value of the function. In case of an error in your R code, R prints a helpful error message for you:

```
> source("fibonacci.R")  
Error in source("fibonacci.R") :  
  fibonacci.R:5:16: unexpected numeric constant  
4:  {  
5:   return 0  
     ^
```

The R code that we can use for finding the factorial of an integer is as follows:

```
myFactorial = function(i)  
{  
  if ( i == 0 )  
  {  
    return(1)  
  }  
  if ( i < 0 )  
  {  
    return(-1)  
  }
```

Quick tip

Probability 101: If you have a six-sided die, the probability of guessing the right number is 1/6. Roll the dice twice, and the probability of guessing right falls to $(1/6) \times (1/6)$ which is 1/36.

Statistical definitions

The following statistical definitions should help you to understand what the output of the **summary()** command means:

- » **Min.** The minimum value in the data set.
- » **Median** The median is an element that divides the data set into two subsets (the left and right subsets) with the same number of elements.
- » **1st Qu.** The first quartile (Q_1) is a value with the property that 25% of the data values in the set are smaller than Q_1 and 75% are larger.

Simplistically, you can consider it as the median value of the left subset of the sorted data set. Note that Q_1 is not necessarily itself a value that appears in the data set.

» **Mean** The mean value of the data set is the sum of all values divided by the number of the items in the data set.

» **3rd Qu.** The third quartile (Q_3) is a value with the property that 75% of the data values in the set are smaller than Q_3 and 25% are larger.

Simplistically, you can consider it as the median value of the right subset of the sorted data set. Again, note that Q_3 is not necessarily itself a value that appears in the data set.

Also note that quartiles can be calculated in different ways, and there is no universal agreement as to the best practice. If you try another statistical package, you may get different results for the values of Q_1 and Q_3 .

» **Max.** The maximum value in the data set.

```
result = 1
for (k in 1:i)
{
  result = k*result
}
return(result)
```

This time the implementation is a little different as it uses a **for** loop instead of recursion. The **for** loop uses a slightly different syntax compared to other programming languages but it is easy to understand and remember.

Creating your own packages

Next, let's organise the code by putting it into an R package. Packages are a good way of organising your own code.

Functions in packages can have any name you want, so long as that name is unique, but it's still good not to use overlapping function names. If they do overlap between different packages, use the package name in front of the function name in order to call the function. For example, **LinuxFormat::function()** instead of just **function()**.

The steps for generating an R package that we want to call **LinuxFormat** and that will include the **myFibo()** and **myFactorial()** functions are as follows:

```
> ls()
character(0)
> source("          # pressing Tab
fibonacci.R factorial.R sort.R
> source("factorial.R")
> source("fibonacci.R")
> ls()
[1] "myFactorial" "myFibo"
> package.skeleton("LinuxFormat")
Creating directories ...
Creating DESCRIPTION ...
Creating NAMESPACE ...
Creating Read-and-delete-me ...
Saving functions and data ...
Making help files ...
Done.
```

Further steps are described in `'./LinuxFormat/Read-and-delete-me'`.

The last R command creates a new directory called **LinuxFormat** – the same name as that of the R package – and you can peruse the contents with:

```
$ ls -lR LinuxFormat/
LinuxFormat/:
```

```
total 20
-rw-r--r-- 1 mtsouk mtsouk 284 Nov  4 10:18 DESCRIPTION
drwxr-xr-x 2 mtsouk mtsouk 4096 Nov  4 10:18 man
-rw-r--r-- 1 mtsouk mtsouk  31 Nov  4 10:18 NAMESPACE
...
```

The **LinuxFormat** package will automatically have the two functions in it, because of the two **source()** calls. You will need to install the package as root so anyone can use it on your Linux system:

```
# R CMD INSTALL LinuxFormat
* installing to library '/usr/local/lib/R/site-library'
* installing *source* package 'LinuxFormat' ...
** R
...
* DONE (LinuxFormat)
```

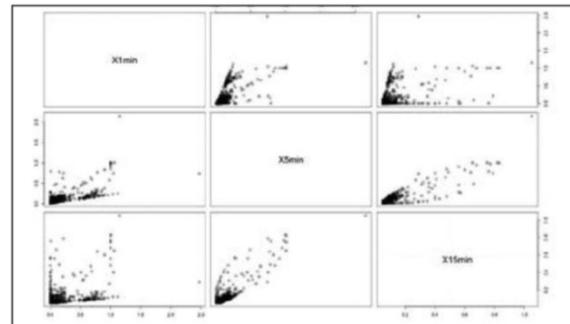
The following code and its output is a proof that the package was successfully installed:

```
# ls -l /usr/local/lib/R/site-library
total 4
drwxr-xr-x 6 root staff 4096 Nov  4 10:33 LinuxFormat
```

Warning: before installing the package you *must* edit both **myFactorial.Rd** and **myFibo.Rd** files and fill in the **\title** fields. If you don't do this, you will get an error message and the installation will fail.

From now on, you can use the new package as follows:

```
> require(LinuxFormat)
Loading required package: LinuxFormat
> ls(getNamespace("LinuxFormat"))
[1] "myFactorial" "myFibo"
> ls()
character(0)
> myFibo(12)
[1] 144
> LinuxFormat::myFibo(12)
[1] 144
```



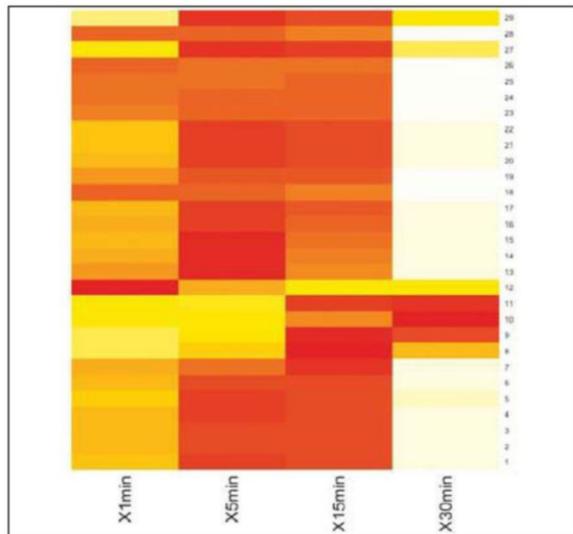
Quick tip

Probability 102:
Following on from Probability 101 (see p88), if you roll two dice at once, then the probability of guessing both numbers won't be the same as before, because you won't have to guess the order that the two numbers will come up.

» **Output from the `pairs()` command (discussed overleaf).**
The R package **ggplot2** uses **ggpairs()**, which improves the output.

» **If you missed last issue** Call 0844 848 2852 or +44 1604 251045

► A heat map is a good and visually impressive way to present your data. Heat maps are great for analysing latency and utilisation monitoring data.



» Next, let's look at how to use R to analyse real data. I'm going to be analysing a set of system-monitoring data I collected, but you could use your own data set.

The **pairs()** command offers a very handy way of finding relations between variables. If you use **ggplot2** – a powerful R package for generating graphics that deserves an article in its own right – you can also use **ggpairs()**, an improved version of **pairs()** that calculates and adds the coefficient of correlation in the output. This is a statistical term used to describe the strength of the relationship between two variables. In simple terms, the closer the value of the coefficient of correlation is to 0, the weaker the relationship between the two variables – that is, the closer they are to being uncorrelated. The closer the value of the coefficient is to +1 or -1, the stronger the correlation is between the variables. A positive coefficient of correlation shows that if one variable increases, the other variable tends to increase as well. A negative coefficient indicates that if one variable increases, the other one tends to decrease.

You can generate output from **pairs()** and **ggpairs()** with the following commands:

```
> data <- read.table("uptime.data", header=TRUE)
> pairs(data)
> require(ggplot2)
> require(GGally)
> require(CCA)
> ggpairs(data)
```

In the image (on p89), you can see the output of the **pairs()** command applied to my system-monitoring data. It shows that the variables X5min and X15min are 'more related' than variables X1min and X15min. In other words, the load average values of a Linux system change more drastically per minute than per five minutes or per fifteen minutes.

Let's save the graphical output generated by R into a new file named **filename.png**. First, you need to open a device using **png()**, **bmp()** or **pdf()**. Then you plot what you want, using the commands you want. Finally, if you're using R remotely, you close the device – note that the final command below is not necessary in R scripts:

```
png(filename="filename.png")
# You now execute the plotting commands you want
dev.off()
```

Quick tip

Statistics can be misleading. If a man is hit by a car going an average speed of 5 kilometres per hour for the last 3 hours, how fast was the car going when the man was hit? Answer: 100 kilometres per hour. Never confuse mean and maximum values.

Next, let's generate a box plot. This is a good way of showing the distribution, variation and median value of a data set at a glance. The top and bottom of the box represent the first and third quartiles of the data set, while the horizontal line in the middle shows the median. The 'whiskers' projecting above and below the box indicate variability outside the quartiles, while circles above or below the whiskers themselves indicate outlying data values.

Box plots excel in visualising metrics, such as visitor time on a page and time to serve a page. As an example, the plot shown at the foot of the page uses multiple samples of the three load average values, taken from the **uptime** command. It was generated using the following R commands:

```
> data <- read.table("uptime.data", header=TRUE)
> boxplot(data, ylab="Uptime Value", xlab="Sample values",
  col="lightblue", border="blue", main="Box Plot of Load
  Averages")
> grid()
```

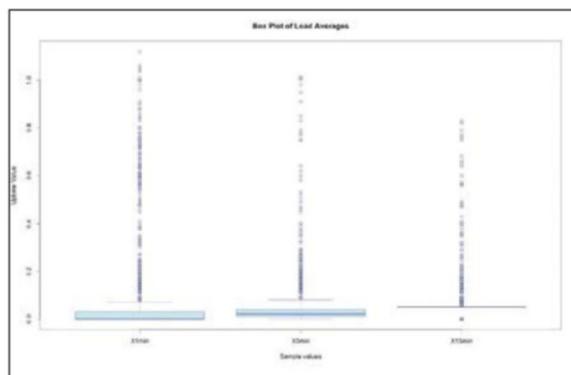
The first command reads data from an external file and saves it to a new variable called **data**. The second command generates the box plot using the **data** set of values. The last command draws a grid on screen for beautifying the output.

Heat maps

A heat map is a way of visualising a table of numbers in which you substitute the real values with coloured cells. They're useful for finding highs and lows, and maybe patterns. Heat maps are best for relatively small data sets. Don't try to use them to visualise more than 500 values or so, as this requires a more detailed knowledge of R.

Monitoring data for multiple computers is a good candidate for a heat map. The map shown on the top right of the page can be generated from my data set using the following commands:

```
> data <- read.table("mapData", header=TRUE)
> data_matrix <- data.matrix(data)
> head(data)
X1min X5min X15min X30min
1 0.5 0.01 0.05 1.1
2 0.5 0.01 0.05 1.3
3 0.5 0.01 0.05 1.3
4 0.5 0.01 0.05 1.2
5 0.5 0.03 0.05 0.9
6 0.5 0.01 0.05 1.3
> heatmap(data_matrix, col = heat.colors(32), Rowv=NA,
  Colv=NA, margins=c(7,10))
```



► A box plot is a good way of showing the distribution, variation and median value of a data set at a glance.

» **Never miss another issue** Subscribe to the #1 source for Linux on page 32.

To draw a heat map using different colours, use **cm.colors**, **topo.colors** or **terrain.colors** instead of **heat.colors**.

Automation and data sorting using R

In my previous article [See Tutorials, p70, LXF192], I explained how to extract monitoring data as text files and process them manually using R. This time, let's create R scripts to automate the process.

First, let's generate heat maps. Each generated image file will have a unique name in order to keep historical data.

The script file, **heatmap.R**, is as follows:

```
#!/usr/bin/env Rscript
now <- format(Sys.time(), "%b%d%H%M%S")
file_base <- "heatMap"
outputfile <- paste(file_base, "-", now, ".png", sep="")
data <- read.table("mapData", header=TRUE)
png(filename=outputfile, width=1280, height=800)
data_matrix <- data.matrix(data)
heatmap(data_matrix, col = heat.colors(32), Rowv=NA,
Colv=NA, margins=c(7,10))
```

Rscript is a front-end for scripting with R and is very handy for running R code using *cron*. If you make **heatmap.R** an executable file, like you would do with a *Bash* script (**chmod 755**), you can run it as a *cron* job without any problems!

We can also implement the famous Bubble sort algorithm in R. The code below is relatively slow but it's easy to understand even if you're not familiar with sorting:

```
mySort = function(set)
{
  len = length(set)
  found = 1
  while (found == 1)
  {
    found = 0
    for (k in (1:(len-1)))
    {
      if (set[k] > set[k+1])
      {
        temp = set[k]
        set[k] = set[k+1]
        set[k+1] = temp
        found = 1
      }
    }
  }
  return(set)
}
```

We use the **(1:(len-1))** shortcut in this implementation, which generates all required loops for bubble sort to work:

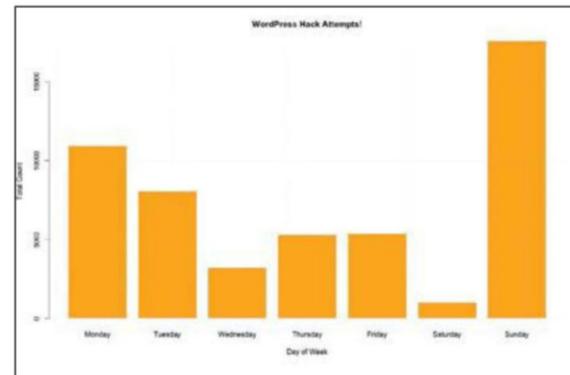
```
> len = 10
> (1:(len-1))
[1] 1 2 3 4 5 6 7 8 9
```

In other words the variable **k** takes all the values from the new set that is created using the **(1:(len-1))** shortcut, one by one. To test the implementation, you can create a test data set with 100 values from 0 to 1,000 using:

```
> test_vec = round(runif(100, 0, 1000))
> mySort(test_vec)
```

The **system.time()** command can help you find out the time it took an operation to finish. It's similar to the Unix **time** command. When sorted, the **system.time()** output should look something like this:

```
> set = c(1, 3, 4, 0, -1)
> mySort(set)
[1] -1 0 1 3 4
```



► A bar plot showing hack attempts on a WordPress site, generated automatically from a log file using R.

```
> system.time(sort(set))
 user  system elapsed
 0.000   0.000  0.001
```

Checking server security using R

Processing log files that contain web server data can be a very demanding job, but R deals with it comfortably!

To import a log file into R:

```
> LOGS = read.table("logfile.log", sep=" ", header=F)
```

As an example, I'm going to analyse a log file from a WordPress site. I want to monitor **POST /wp-login.php**, **HTTP/1.1, POST /wp-login.php HTTP/1.0, GET /wp-login.php HTTP/1.1** and **GET /wp-login.php HTTP/1.0** requests that indicate brute-force hack attempts.

Only columns **V4** and **V6** are of interest to me, so I can isolate them from the **HACK** variable, as follows:

```
> names(LOGS)
[1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9"
"V10"
> HACK = subset(LOGS, V6 %in% c("POST /wp-login.php
HTTP/1.1", "POST /wp-login.php HTTP/1.0", "GET /
wp-login.php HTTP/1.0", "GET /wp-login.php HTTP/1.1"))
> names(HACK)
[1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9"
"V10"
> HACK[1:3] <- list(NULL)
> names(HACK)
[1] "V4" "V5" "V6" "V7" "V8" "V9" "V10"
> HACK$V5 <- NULL
> HACK[3:5] <- list(NULL)
> HACK[3:4] <- list(NULL)
> names(HACK)
[1] "V4" "V6"
```

Next, I can extract the day of the week from column **V4** and generate a bar plot:

```
> newV4 <- strptime(HACK$V4 ,
format(['%d/%b/%Y:%H:%M:%S'])
> day = format(newV4, "%A")
> barplot(table(factor(day, levels=c("Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday",
"Sunday"))), xlab="Day of Week", ylab="Total Count",
col="orange", border="lightblue", main="WordPress Hack
Attempts!")
> grid()
```

As you might expect, the plot (shown at the top of the page) indicates that most hack attempts happen on Sundays, when the system is not being monitored.

That concludes our exploration of R and its use in analysing system data. Always remember that a heat map or a histogram is just another drawing: your own data will give meaning to every plot you create. **LXF**

Answers

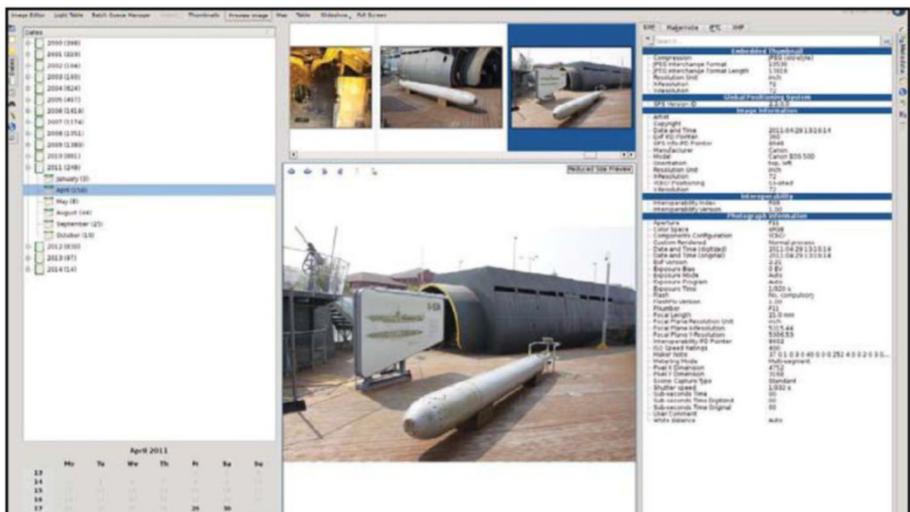
Got a question about open source? Whatever your level, email it to lxf.answers@futurenet.com for a solution.

This month we answer questions on:

- 1 Sorting photographs
- 2 Securely wiping USB sticks
- 3 Dealing with 'foreign' disks
- 4 Cloud security and encryption
- 5 Copying MP3 files in order
- 6 Resolving btrfs problems

1 Sorting photographs

QI've got over 40 directories containing a couple of thousand individual photographs that I'd really like to sort, and I'd like to sort them by date, preferably by their creation date. But as I understand it, Linux doesn't currently have a creation date, only a modification date. Is there any way I copy them into one big directory, run a sort command, then have the photos in this directory shown visually in their correct order in a GUI, so that I can then put them back into individual directories based upon



Thanks to EXIF tags, *digiKam* can display photos in virtual date folders spread across disks.

their timestamp? I've done a bit of searching online, but I've found that there's nothing immediately obvious. Any help would be gratefully appreciated.

GeordieJedi
From the forums

ADigital camera files contain data other than the actual image, called EXIF (see <http://bit.ly/EXIFDefined> for more detail). This can include exposure settings; lens focal length; ISO value; whether flash was used; and even GPS co-ordinates. Most importantly, EXIF data includes a timestamp. Many digital image-handling programs can work with this information. *DigiKam*, my personal choice, displays this information with each thumbnail, and shows more when you select an image. It also enables you to view photos by date, presenting them in virtual year and month folders and sub-folders, whatever their on-disk organisation.

If you want to re-organise thousands of files like this, you're going to have to do some scripting and the program you need is *ExifTool*. This reads (and writes, but that's not relevant here) EXIF tags. Using the following command:

`exiftool photo.jpg`

shows you all the EXIF data for a file, but we are

only interested in one tag, `DateTimeOriginal`,

which you can read with:

`exiftool -DateTimeOriginal photo.jpg`

The output from *ExifTool* can be customised

however you want. Read the man page for

details of the `-p` and `-d` options, but this will

give the sort of information you need:

`exiftool -d '%Y %m' -p '$FileName`

`$DateTimeOriginal' *.jpg`

For each photo, it outputs three items on a line: the name of the file, the year it was taken and the month. Now you can use a short script to move all the files:

Enter our competition



if (\$letter == winner)
get \$books

Get your questions answered and exploit our generosity.



Linux Format is proud to produce the biggest and best magazine about Linux and free

software that we can. A word count of LXF185 showed it had 67,790 words: about as much as *Heart of Darkness* and *Animal Farm* combined, but with way more Linux, coding and free software!

That's more than most of our competitors, and that's how we know that we're big. As for the best, well... that's a subjective claim, but it's one we're happy to stand by.

Because we're nice chaps as well as Linux gurus, and because our offices have very little storage space, we're giving one lucky reader each issue the chance to win some of the books that get sent to us for review. They're mostly reference books that will give your shelves an air of computational gravitas. For a chance to win one, email your question to

lxf.answers@futurenet.com, or post it to www.linuxformat.com/forums to seek help from our lively community of readers.

See page 94 for our star question.

Terminals and superusers

We often give a solution as commands to type in a terminal. While it is usually possible to do the same with a distro's graphical tools, the differences between these mean that such solutions are very specific. The terminal commands are more flexible and, most importantly, can be used with all distributions.

System configuration commands often have to be run as the superuser, often called root. There are two main ways of doing this, depending on your distro. Many, especially Ubuntu and its derivatives, prefix the command with **sudo**, which asks for the user password and sets up root privileges for the duration of the command only. Other distros use **su**, which requires the root password and gives full root access until you type **logout**. If your distro uses **su**, run this once and then run any given commands without the preceding **sudo**.

```
exiftool -d '%Y %m' -p '$FileName
$DateTimeOriginal' *.jpg | while read NAME
YEAR MONTH; do
mkdir -p $YEAR/$MONTH
mv -i $NAME $YEAR/$MONTH
done
```

Using **-i** with **mv** means it will stop and prompt you if the filename already exists in the destination, protecting you against the sort of unfortunate typo that tries to copy all your photos to the same file name. Instead of using a wildcard, you can give a directory name to process all suitable files in that directory, or add **-r** to descend into subdirectories too. You can process your entire photo collection with something like:

```
exiftool -d '%Y %m' -p '$FileName
$DateTimeOriginal' -r ~/Photos
```

Just make sure you move the files into a different directory, or you will be trying to rename them over themselves.

2 USB from nothing

QA few weeks ago, a team of German researchers demonstrated that USB keys are not safe, even after reformatting! I suppose that SD cards are also affected.

A possible solution could be to zero the USB stick with:

sudo cat /dev/zero >/dev/sdb
where **sdb** is the USB stick. Can you confirm this is indeed an effective solution to wipe out any malware?

The problem is that after being 'zeroed', a USB stick is no longer readable under Linux. (Windows still can 'see' it, and offers the option to reformat it – why can't Linux do that?) If it's possible, how can I reformat the 'zeroed' USB stick? And after formatting, how do I mount the USB stick?

Oliver

ANo device is safe after formatting as the act of formatting removes very little data from the drive, if any. Filling a device with zeroes is a reasonable way of overwriting all previous data. However, you zeroed the whole device, not just the filesystem, by writing to **/dev/sdb** instead of **/dev/sdb1**. That means you also erased the drive's partition table. It's possible to format a USB flash drive without partitioning, although it seems more commonplace on Windows. There's no particular point in partitioning a USB stick, as you can just format the whole device, which you can do from the CLI with:

sudo mkfs.vfat -I -F 32 /dev/sdb

The **-I** option is needed because the Linux **mkfs** tools will not normally create a filesystem on an unpartitioned device (except for a floppy disk, if you remember those). Generally, running **mkfs** on a whole disk is likely to be a mistake that could wipe out your whole drive, so it's not allowed by default; using **-I** overrides this safeguard. If you want to partition the device, you will need to use **fdisk** to create a single partition filling the whole space and then run the above command without the **-I** and with **sdb1** instead of **sdb**.

However, you would probably find it easier to use **GParted** to do the whole partition and format job in one go. If your distro doesn't have **GParted** installed by default, it will be in its software library. Note that you need **GParted**,

not **parted**. The latter is a CLI tool that's even less friendly to the new user than **fdisk**.

Most desktops include automounters for removable devices, so after formatting the USB stick, unplug it and plug it back in to have it recognised and mounted. If you want to mount it from the command line, the syntax is:

mount /dev/sdb (or sdb1) /media/usbstick
where **/media/usbstick** is where you want the contents of the device to appear; this directory must already exist.

3 Long-lived Atari

QI have a mid-1980s Atari ST for which I have an external HDD unit, consisting of a SCSI hard drive with 50-way connector, plus a 'host adaptor' which forms the bridge between the standard SCSI interface on the drive and the strange Atari version of SCSI (ACSI).

I want to verify that the HDD is working, and if it is, image it to another drive to back it up. I have an Adaptec PC PCI SCSI interface on the way to me for just this purpose. The idea is to use Linux to image the SCSI drive to an image file on the Linux machine's hard drive or a USB stick. The capacity of the Atari HDD is about 20MB so I could save umpteen backup files on even the most modest of external devices.

The only area of concern that I have is possible corruption of the Atari HDD if my Linux system of choice tries to automount it. I'm concerned that this may happen even with live-boot CDs, which boot to a desktop complete with file browser and so on. Am I worrying about nothing? Or should I use a distro which boots straight to a terminal?

This drive contains 30-plus years of irreplaceable material, so you'll understand why I'm nervous about the possibility of damaging it. Is the file system (Atari ST GEMDOS, probably) likely to be supported by Linux?

SiriusHardware
From the forums

after 10 seconds, but requires that the system gets as far as mounting the root filesystem to read this file. More robustly, you can add **panic=10** to the kernel parameters in your **Grub** menu.

You may wonder about the point of forcing a reboot if the kernel will only panic again and reboot again, ad infinitum. The answer lies in **Grub**'s ability to set up a fallback boot. This is a godsend when trying a new kernel on a remote computer – far less embarrassing than having to phone and ask someone to reboot because you messed up a kernel config.

A quick reference to...

Kernel panic

Occasionally, the Linux kernel hits a situation it just cannot handle and, ignoring the advice on the cover of the greatest book ever written, it panics. This manifests itself as a cryptic message to the console (or hidden behind a pretty boot splash screen if you are really unlucky) and a blinking caps lock LED. If it has got as far as mounting the root filesystem read/write, it can also write diagnostic information to the disk: a 'core dump'.

In theory, you should never see one of these using the kernel your distro provides. If you are using a hand-rolled kernel, it usually means you've forgotten to build in support for the disk drive controller and filesystem used by the root filesystem, meaning it cannot be mounted.

You'd be forgiven for thinking that once this happens, all is lost, but the kernel can still manage one last act of dignity, even if you kill it like this. The kernel can be told to reboot after a panic. Adding **kernel.panic=10** to **/etc/sysctl.conf** will reboot it

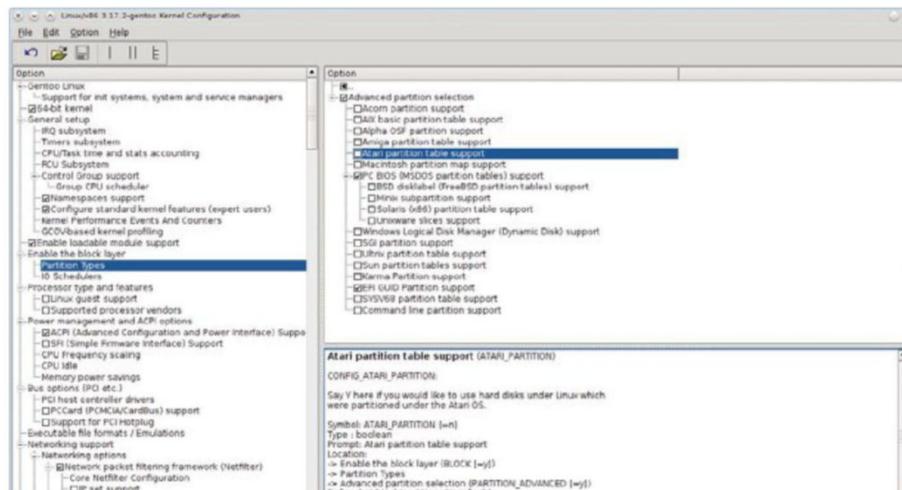
Answers

A Automounters are fairly benign programs. They attempt to mount the filesystem and give up if it doesn't work. If the system can't mount the filesystem, it can't write to it, so you are safe. They are also not limited to desktop usage, so booting to a terminal may still try to mount it. I'd recommend the use of SystemRescueCd. It does try to mount all attached devices when it boots, in order to look for its **sysrcd.dat** file, but gives up if the partition does not contain a 'standard' filesystem. To be extra safe, you can use the **skipmount** boot option to have it ignore one or more partitions. You will need to head into the BIOS menu first to check the drive order, then press Tab on the boot menu and add, for example:

```
skipmount=/dev/sdb1 skipmount=/dev/sdb2
```

Note that you specify individual partitions, not whole disks. Once you are booted into SystemRescueCd, you can use **dd** to copy the data. As **dd** reads the raw data from the drive directly, it doesn't care about partitions or filesystems.

The Linux kernel does have options for Atari partition tables and filesystems, so you could conceivably mount the drive and copy off the data you need – although I would advise backing up with **dd** first. However, you may have trouble finding a live CD with a kernel, which has the **ATARI_PARTITION** option (SystemRescueCd's kernel doesn't). Incidentally, the 50-pin connector was the standard for SCSI. I think I still have some SCSI ribbon cables knocking around – not thrown out 'just in case'.



Linux is able to read disk layouts and filesystems for many old (and new) systems – even if you were unfortunate enough to own an Atari instead of an Amiga!

4 Cast-iron cloud

Q Currently, I am using Dropbox as platform-independent cloud storage, but according to Snowden that's not good. He recommends SpiderOak.

After doing some research, I found the following platform-independent (Microsoft, Apple, Linux, iOS, Android), secure cloud storage providers: SpiderOak (2GB, free), Wuala (5GB, \$0.99/month), Mega (50GB, free) and Tresorit (5GB, free). Which one do you recommend and why?

Or I could stick with using Dropbox for now, but start encrypting my files. For encrypting files which program do you recommend: **TrueCrypt** or **7-Zip**? It needs to

be platform-independent. Also, which platform-independent password manager do you recommend?

JohanM

A There is one option noticeable by its absence from your list: **OwnCloud**. If you want security, run **OwnCloud** locally or get yourself a VPS and install it there. There was a tutorial on setting up **OwnCloud** in a previous issue [see *Tutorials*, p70, **LXF190**].

Any system that uses encrypted storage requires someone to know the encryption keys. SpiderOak is more secure than Dropbox because the encryption is done locally using keys stored only on your system. Other systems may allow someone with the required

Star Question Winner!

This month's winner is Neal Davidson. Get in touch with us to claim your glittering prize!

Broken btrfs

Q I've been using btrfs for a while but recently it started to play up. Shortly after booting all my btrfs sub-volumes become read-only. When I reboot, all is well for a while and it starts again. I cannot find anything in **/var/log/messages** but **dmesg** returns lines like this:

```
[ 291.245123] parent transid verify failed on  
11234344960 wanted 247156 found 250750  
[ 291.246245] parent transid verify failed on  
159174590464 wanted 247379 found 250606  
[ 291.246814] parent transid verify failed on  
159174590464 wanted 247379 found 250606  
followed by a stack trace that mentions  
btrfs. Running fsck fixes nothing either.
```

Neal Davidson

A Btrfs is a good filesystem and could well become the default for many distros in the future, but it's still under heavy development, although considered stable. This means that you should always use the latest kernel available for your distro and

keep regular backups. The errors you are seeing are due to an inconsistency between the cache that btrfs uses and the actual filesystem, possibly caused by a hardware error or a power failure. As with some other filesystems, when btrfs detects an error it tries to protect against further corruption by remounting itself as read-only.

The first step to take is to remount the filesystem with the options:

```
recovery,nospace_cache,clear_cache
```

If this is your root filesystem, add the options to **fstab** and reboot. The recovery option forces Btrfs to try to fix and errors; the other options cause it to ignore the cache. The recovery may take a while. This is the only non-destructive option, so if it doesn't work, the next step is to back up your data with:

```
btrfs restore DEVICE-NODE BACKUP-DIRECTORY
```

using the appropriate values for the device and destination, which should be on a different drive formatted with a Linux filesystem. If you

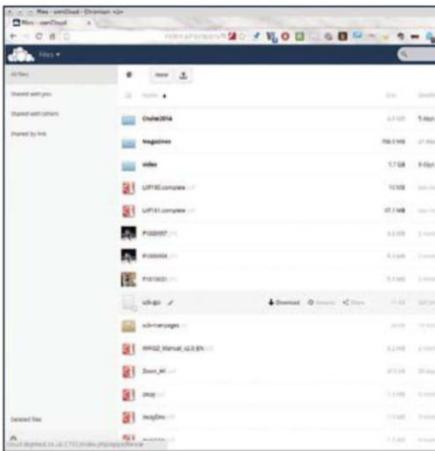
want to back up your snapshots too, add the **-s** option. Use a live CD, as this has to be done with the filesystem unmounted, and use a recent kernel. I would use SystemRescueCd with the alternate kernel boot option. Now, still running from the live CD, try clearing the filesystem's log with:

```
btrfs-zero-log DEVICE-NODE
```

Reboot and see if this has cured your problem. If not, the final option is to **fsck** the drive from live CD, not with **fsck.btrfs**, but with the main btrfs program:

```
btrfs check --repair DEVICE-NODE
```

This will take a long while to run but will check and repair your drive. In cases of serious corruption, you may need to add one of **--init-extent-tree** or **--init-csum-tree**, but if your filesystem were that corrupted, it probably wouldn't mount in the first place. You may lose some data with the **btrfs-zero-log** or **--repair** operations – some files may have reverted to older versions – but hopefully the loss will be minimal.



► If you don't trust the commercial and free cloud providers, you could always run your own with *OwnCloud*.

privileges access to your data – whether that's a rogue employee selling it on or someone acting in accordance with the wishes of a government agency. The disadvantage to that approach is that you can only access your data from a device with your keys on it.

Security is not only about the secrecy of your data but also its safety. *OwnCloud* gives you full control but also all the responsibility of keeping your system safe from hardware or software failures. Large providers, like the ones you mentioned, have good backup and redundancy, but you get what you pay for.

As for TrueCrypt, it's no longer officially supported. If you want to encrypt individual files or archives, use GPG or PGP as they are

available on all platforms. For password management, I use *KeePass* (<http://keepass.info>). It has clients for desktop and mobile platforms and stores everything in a single encrypted file, making it easy to store in the cloud safely, even on Dropbox.

5 MP3 to ABC

QI have some music in a directory which contains sub-directories for artists and albums. When I copy this music to a USB flash drive for use in my car, I want it to be copied in the order that I see it ordered on my computer, which is alphabetically. However, it seems to copy the MP3 files in, I guess, the disk's order? I want to play my music in the order I want, not some new variation on 'shuffle'.

nuxguy

AThis is a common problem with some basic MP3 players. Files are saved to a disk or flash drive in no particular order (as they say on reality TV vote-offs) and leave it to the software creating a directory listing to do the sorting. This is a faster and more efficient way of using the disk.

Copying in alphabetical order may help, although there's still no guarantee that files will be added to the drive's FAT list in the order you want. Even if it did, what happens when you want to add more files? You would need to reformat the drive and copy everything again, just to add one more song or album.

This is a common enough problem that there is a program to deal with it called *FATSort* (<http://fatsort.sourceforge.net>). Run it after

Help us to help you

We receive several questions each month that we are unable to answer because they give insufficient detail about the problem. In order to give the best answers to your questions, we need to know as much as possible.

If you get an error message, please tell us the exact message and precisely what you did to invoke it. If you have a hardware problem, let us know about the hardware. If Linux is already running, you can use the *Hardinfo* program (<http://sourceforge.net/projects/hardinfo.berlios>) that gives a full report on your hardware and system as an HTML file you can send us. Alternatively, the output from *lshw* is just as useful (<http://ezix.org/project/wiki/HardwareLister>).

One or both of these should be in your distro's repositories. If you are unwilling, or unable, to install these, run the following commands in a root terminal and attach the *system.txt* file to your email. This will still be a great help in diagnosing your problem.

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

copying any files to the USB stick and it will rearrange the FAT listing so that files appear in alphabetical order:

```
fatsort -c /dev/sdb1
```

Note: the USB stick should not be mounted when you run *FATSort*, otherwise you could corrupt the filesystem. The **-c** option enables case-insensitive sorting, which you probably want. There are several options described in the *FATSort* man page. For instance, **-l** is useful as it shows the current order of files on the device. You can also see the order of files on a mounted device using **ls**. By default, **ls** sorts output alphabetically, but you can disable this and see the natural disk order with **ls -U**.

Frequently asked questions...

Printing

► What is CUPS?

The Common Unix Printing System, CUPS is a set of drivers and utilities that provide full printer support, management and usage tools for Linux and other Unix-based operating systems.

► So it's a printer driver?

Well yes, but much more than that too. CUPS provides a 'portable printing layer' between your applications and your printing hardware. Naturally, this includes drivers, but it also provides everything else programs need to be able to print.

► Does that mean I have to use the command line and edit configuration files?

Not at all. CUPS provides its own graphical configuration tools that

work through a web browser. Point your favourite browser to <http://localhost:631> (you may need to give your user or root password) and you'll see the CUPS home page. From here you can list, add and remove printers, as well as manage print queues (pending print tasks) and read the documentation.

► Why do I have to use a browser instead of a more standard graphical program?

A web interface means you don't need any graphical toolkits installed; you don't even need X running on the computer. You could access the interface through a text browser like *Elinks*, or from a browser on another computer.

► Isn't that a bit insecure?

It could be, but the default CUPS settings only allow connections from **localhost**. You can alter this

to allow connections for your LAN (allowing internet access is not generally recommended) and also control which users can access which parts of the configuration – you may want to allow users to manage their own print jobs but not delete your printers.

► Where does Gimp-Print fit in with all this, and what has the Gimp to do with printing?

Gimp-Print is a set of printer drivers originally developed for use with the *Gimp*. Although *Gimp* works well with CUPS, some printers may give better output with the Gimp-Print drivers.

These drivers also work with CUPS now, so you can consider Gimp-Print to be an extended set of drivers that works with all programs printing through CUPS, not just *Gimp*.

The project is now called Gutenprint, to avoid this confusion,

but some distros still have Gimp-Print packages.

► What about Gutenprint?

That's the new name for Gimp-Print. Depending on how bleeding-edge a distro you use, you will have Gimp-Print 4.x or Gutenprint 5.x. Now that Gimp-Print has far more to do with printing than *Gimp*, the new name is less misleading, although any name change can cause short-term confusion, and this is already a confusing area.

► How do I find out if my printer is supported?

Your first port of call should be www.linuxprinting.org, which maintains a database of how well (or not) each printer is supported and advice on which drivers to use. Ideally, you should visit this before buying a printer, rather than finding out that your new purchase is considered a paperweight.

On the disc

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

The best of the internet, crammed into a phantom-zone-like 4GB DVD.

Bit bias

All of the distros on the DVD



month this month are 64-bit.

This is probably the first time ever we have produced a DVD without a single 32-bit distro.

This is not part of some great conspiracy to alienate users of (much) older hardware; it is simply the way things are progressing and how they turned out this month. We will continue to include 32-bit distros on the DVDs where appropriate, but we expect the number to decrease.

We are already seeing distros that release only 64-bit versions, such as XBMCbuntu. That makes sense; a ten-year-old PC is hardly the ideal choice to play HD multimedia. As desktops gain more features and benefit from more memory and faster processors, the 32-bit versions of many distros will become less relevant.

That's not to say there isn't a place for 32-bit distros, either in the Linux ecosystem or on the *Linux Format* DVDs. There are several distros that are ideal for, even intended for, older hardware. They will continue to release 32-bit versions and we will continue to feature them. But we need to accept that it's almost 2015 and we have to say goodbye to some of the older systems, even those beloved Amigas...

Neil

» Important NOTICE!

Defective discs

In the unlikely event of your *Linux Format* coverdisc being in any way defective, please visit our support site at www.linuxformat.com/dvdsupport for further assistance. If you would prefer to talk to a member of our reader support team, email us at discsupport@futurenet.com or telephone +44 (0) 1225 822743.

The film buff's friend

64-bit

XBMCbuntu 13.2

This month's feature is about XBMC, so we thought we would give you the chance to try it easily.

XBMCbuntu is actually based on Lubuntu with XBMC pre-installed. When you boot the live environment, it goes straight to XBMC. If you want to install it, log out of XBMC and select XBMCbuntu

from the menu at the top-right of the screen before logging back in as **xbmc** with no password. This gives you the LXDE desktop, from where you can run the installer. XBMC will refuse to load if your graphics card or driver is not suitable. In this case, you will only see the login screen.



Early supporter of btrfs

64-bit

OpenSUSE 13.2

We think of Ubuntu as a long-established distro but OpenSUSE has a history going back almost twice as far. Originally based on Slackware, it soon switched to RPM and became an independent distro in its own right. As with Red Hat, SUSE has

two main lines. The enterprise distro is SLES (SUSE Linux Enterprise Server) while the community desktop project is OpenSUSE. SUSE has long been a supporter of the KDE desktop and that's the version we have on the DVD, but there

are plenty of other desktops to choose from and install once you have installed the distro. One of SUSE's most notable features is YaST, which stands for Yet another Setup Tool but is much more than that. YaST is a fully integrated administration interface that handles everything from initial installation to package management, networking to hardware configuration. And there's a lot more to OpenSUSE than YaST: it's a complete and mature distro that is not afraid to try new technologies, being the first major distro to use btrfs as its default filesystem.





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



**Are you reading
this on a tablet?**
Download your DVD from
www.archive.org

Canonical's latest spin

64-bit

Ubuntu 14.10

Yes, we had Ubuntu 14.10 on a recent disc, but it is relevant to this month's magazine content. Whereas last time around we used one of our remixes with several additional desktops, this is the vanilla (not that vanilla is usually purple) Ubuntu Unity desktop experience. This gives you a quicker and smaller installation, and you can add any other desktops you want to try after installing Ubuntu by searching the package manager for the relevant desktop package, such as **xubuntu-desktop** for Xfce or **kubuntu-desktop** for KDE.



A sleek media centre

64-bit

OpenELEC 4.95.3

OpenELEC is another XBMC-based distro, and this time built from the ground up, so it's much smaller and takes up less disk space – although that's unlikely to be an issue if you're going to need many gigabytes of storage for a media centre: does a few hundred megabytes make that much difference? The distro isn't supplied as an ISO image, so you

will find a TAR file on the DVD. Unpack this to your hard drive, **cd** to the folder containing it, plug in a USB stick and run:

```
sudo ./create_installstick /dev/sdX
```

where **sdX** is your USB drive. If you're not sure which device it is, after plugging it in, run:

```
dmesg | grep 'removable disk'
```



After running **create_installstick** you can boot from that stick to install OpenELEC. If your target system doesn't have a DVD drive, you can run the package on a different PC to create the stick. There's also a Windows BAT file to do the same thing if you're making use of a Windows machine.

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.

Plop 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.

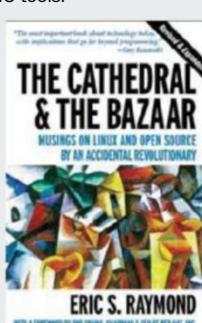
Introduction to Linux A handy guide full of pointers for new Linux users.

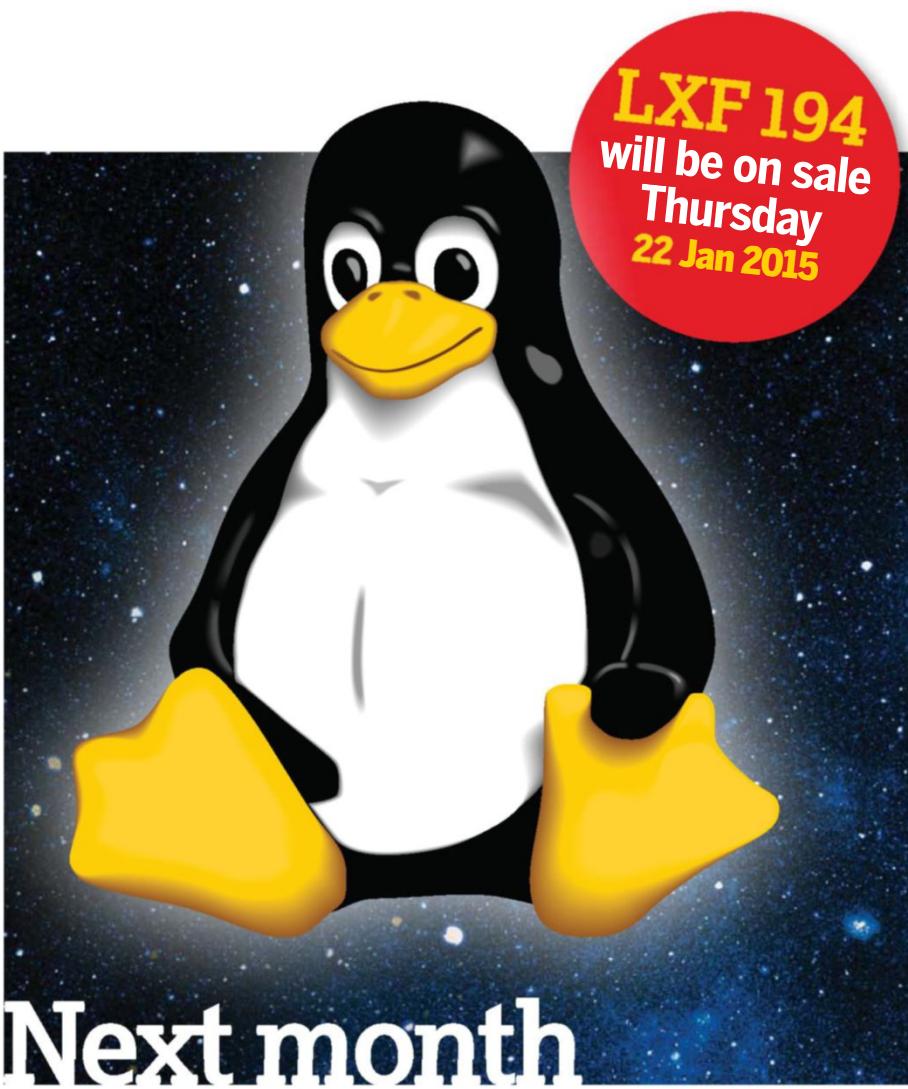
Linux Dictionary The A-Z of everything to do with Linux.

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.





Next month Build a next-gen Linux system

Live life on the cutting edge with our complete guide to next-generation graphics, filesystems and more.

Make mice in Micro Python

Full project: use the microprocessor-friendly version of Python 3 to turn a pyboard into a USB mouse.

The best Linux desktop

Our group test of the best environments that Linux has to offer, from Cinnamon to Enlightenment.

Become a guitar hero!

Channel your inner Santana by using the *guitarix* virtual guitar amplifier to add effects to your solos.

Contents of future issues subject to change – we might have been busted by the storage police.

LINUX
FORMAT
#1 for Free Software

Future Publishing, Quay House,
The Ambury, Bath, BA1 1UA Tel 01225 442244
Email linuxformat@futurenet.com

 20,238 January – December 2013
A member of the Audit Bureau of Circulations.

EDITORIAL

Editor Neil Mohr

neil.mohr@futurenet.com

Technical editor Jonni Bidwell

jonni.bidwell@futurenet.com

Operations editor Chris Thornett

chris.thornett@futurenet.com

Art editor Efrain Hernandez-Mendoza

efrain.hernandez-mendoza@futurenet.com

Editorial contributors Russell Barnes,

Danil Baturin, Neil Bothwick, Chris Brown,

Sean Conway, Matt Hanson, Tom Hatfield, Kevin Lee,

Andrew Mallett, Nick Peers, Les Pounder, Mayank

Sharma, Shashank Sharma, Richard Smedley,

Matt Swider, Alexander Tolstoy, Mihalis Tsoukalos

Editorial support Alex Summersby, Jim Thacker

Art support Alvin Weetman

Illustrations Shane Collinge, Simon Middleweek

ADVERTISING

For ad enquiries please contact:

Richard Hemmings

richard.hemmings@futurenet.com

MARKETING

Marketing manager Richard Stephens

richard.stephens@futurenet.com

PRODUCTION AND DISTRIBUTION

Production controller Marie Quilter

Production manager Mark Constance

Distributed by Seymour Distribution Ltd, 2 East

Poultry Avenue, London EC1A 9PT Tel 020 7429 4000

Overseas distribution by Seymour International

LICENSING

Licensing & syndication manager Regina Erak

regina.erak@futurenet.com Tel +44 (0)1225 442244

Fax +44 (0)1225 732275

CIRCULATION

Trade marketing manager Juliette Winyard

Tel 07551 150 984

SUBSCRIPTIONS & BACK ISSUES

UK reader order line & enquiries 0844 848 2852

Overseas reader order line

& enquiries +44 (0)1604 251045

Online enquiries www.myfavouritemagazines.co.uk

Email linuxformat@myfavouritemagazines.co.uk

MANAGEMENT

Content & Marketing director Nial Ferguson

Head of Content &

Marketing, Technology Nick Merritt

Group editor-in-chief Paul Newman

Group art director Steve Gotobed

Editor-in-chief, Computing Brands Graham Barlow

LINUX is a trademark of Linus Torvalds. GNU/Linux is abbreviated to Linux throughout for brevity. All other trademarks are the property of their respective owners. Where applicable code printed in this magazine is licensed under the GNU GPL v2 or later. See www.gnu.org/copyleft/gpl.html

Copyright © 2014 Future Publishing Ltd. No part of this publication may be reproduced without written permission from our publisher. We assume all letters sent – by email, fax or post – are for publication unless otherwise stated, and reserve the right to edit contributions. All contributions to *Linux Format* are submitted and accepted on the basis of non-exclusive worldwide licence to publish or license others to do so unless otherwise agreed in advance in writing. *Linux Format* recognises all copyright in this issue. Where possible, we have acknowledged the copyright holder. Contact us if we haven't credited your copyright and we will always correct any errors. We cannot be held responsible for mistakes or misprints.

All DVD demos and reader submissions are supplied to us on the assumption they can be incorporated into a future cover-mounted DVD, unless stated to the contrary.

Disclaimer All tips in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your computer, peripherals or software through the use of any tips or advice.

Printed in the UK by William Gibbons on behalf of Future.



Future is an award-winning international media group and leading digital business. We reach more than 49 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-Maddick
Non-executive chairman Peter Allen
Chief financial officer Richard Hale

Tel +44 (0)207 042 4000 (London)
Tel +44 (0)1225 442 244 (Bath)

We are committed to using paper from well-managed, certified forestry and chlorine-free manufacture. Future Publishing and its paper suppliers have been independently certified in accordance with the rules of the FSC (Forest Stewardship Council).

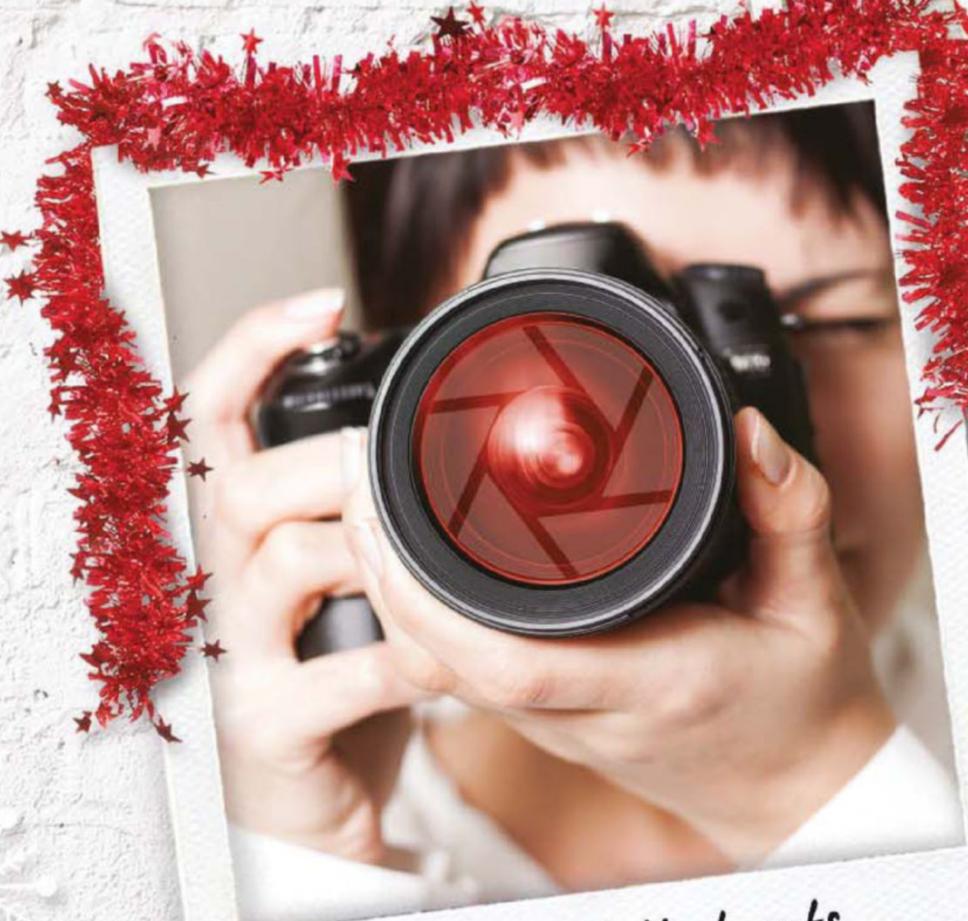


When you have finished with
this magazine please recycle it.



The Photography Show

21-24 MARCH 2015
THE NEC, BIRMINGHAM



The event that puts
you in the picture

**The Photography Show is the event for
enthusiast photographers of all levels.**

With an incredible line-up of leading brands, show bargains, live demos and inspirational speakers, you'll enjoy a great day out dedicated to all things photography.

Tickets to The Photography Show are the perfect gift this Christmas. Book using **TPS15XMSAD** at photographyshow.com and get an exclusive discount today. Merry Christmas!



@ukphotoshow



YOU
CAN
TAKE
HIM
AWAY
FROM
HERE



Sam's surrounded. By bullets. By screams. By bombs and blood.
He's 10.

There are thousands of children like Sam in Syria right now.
They're too young to understand what's happening. But they're
old enough to be traumatised by every horrific thing they see.

Please help us take them to one of War Child's 'Safe Spaces' -
a haven away from the violence. Somewhere they can feel
protected and cared for, where they can just be children and play.

**Text 'PROTECT' to 70444 and give £3
to help protect Syrian children like Sam.**

warchild.org.uk/syria

WAR
child