

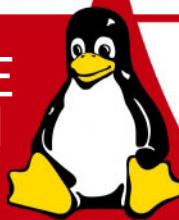
SECURE STORAGE: DISKASHUR PRO 2

www.linuxuser.co.uk

LinuxUser

THE ESSENTIAL MAGAZINE
FOR THE GNU GENERATION

& Developer™



ULTIMATE RESCUE & REPAIR KIT

- Digital forensics
- Data recovery
- File system repair
- Partitioning & cloning
- Security analysis



INTERVIEW

Vivaldi

The web browser for Linux power users



IN-DEPTH GUIDE

The future of programming

The hot languages to learn



PRACTICAL PI

Build an AI assistant
Python & SQLite
Micro robots!



42 PAGES OF GUIDES

- › MQTT: Master the IoT protocol
- › Security: Intercept HTTPS
- › Essential Linux: The Joy of Sed

Pop!_OS

The distro for creators, developers and makers

Get into Arch Linux

4 Linux distributions for entering the world of Arch

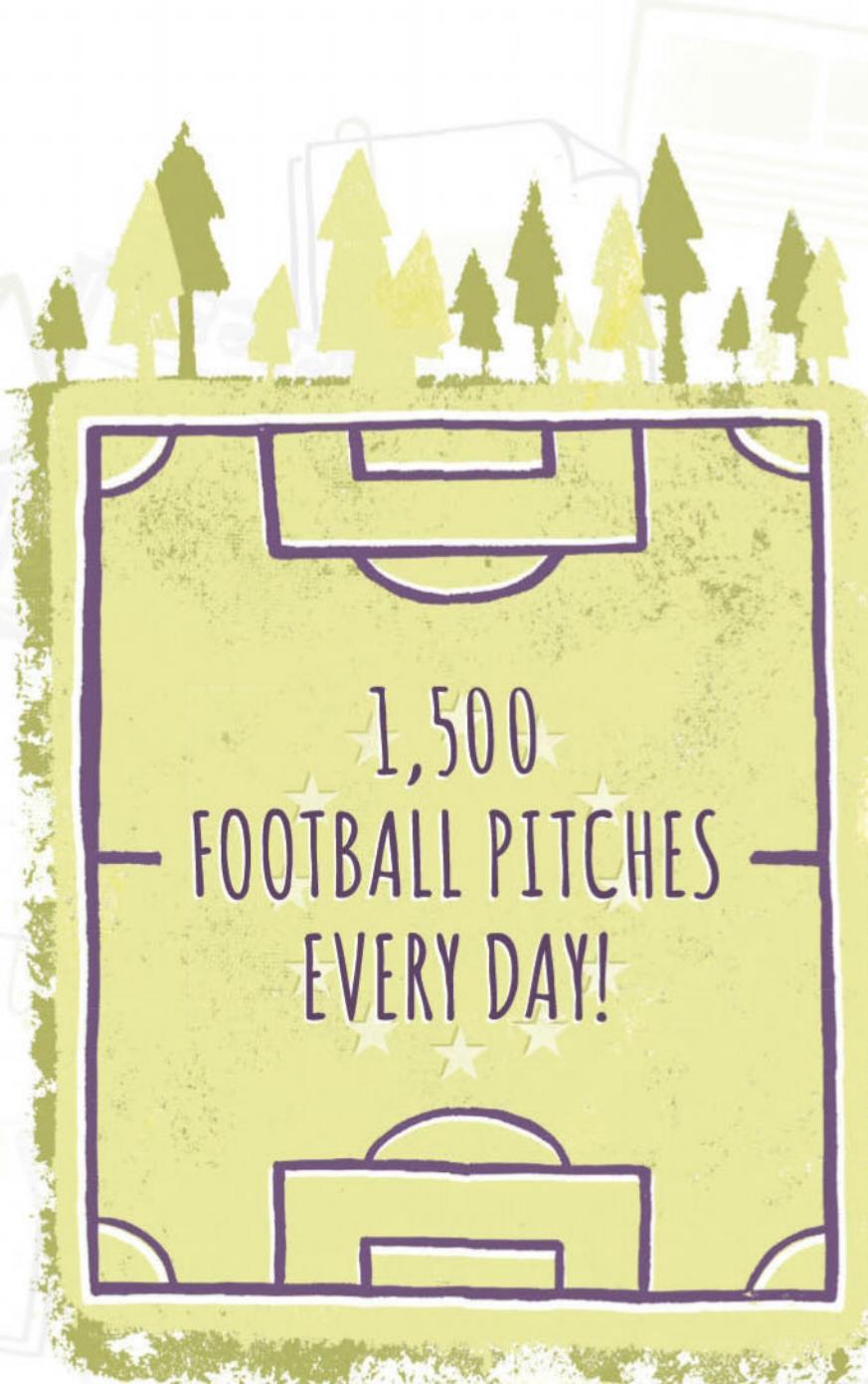


ALSO INSIDE

- » Dev report: Kernel 4.15
- » Java: Spring Framework
- » Top web admin tips



Future



1,500
FOOTBALL PITCHES
EVERY DAY!

Did you know that European forests, which provide wood for making paper and many other products, have grown by 44,000km² over the past 10 years? That's more than 1,500 football pitches every day! [†]

Love magazines? You'll love them even more knowing they're made from natural, renewable and recyclable wood



[†]UNFAO, Global Forest Resources Assessment 2005-2015.

Two Sides is a global initiative promoting the responsible use of print and paper which, when sourced from certified or sustainably managed forests, is a uniquely powerful and natural communications medium.

There are some great reasons to #LovePaper
Discover them now,
twosides.info



Future Publishing Limited
Quay House, The Ambury,
Bath BA1 1UA

Editorial

Editor **Chris Thorne**
chris.thornett@futurenet.com
01202 442244

Designer **Rosie Webber**
Production Editor **Phil King**
Editor in Chief, Tech **Graham Barlow**
Senior Art Editor **Jo Gulliver**

Contributors

Dan Aldred, Michael Bedford, Joey Bernard, Neil Bothwick, Christian Cowley, Nate Drake, John Gowers, Tam Hanna, Toni Castillo Girona, Mel Llaguno, Paul O'Brien, Jon Masters, Katherine Marsh, Calvin Robinson, Mayank Sharma, Alexander Smith, Steve Wright

All copyrights and trademarks are recognised and respected.
Raspberry Pi is a trademark of the Raspberry Pi Foundation.

Advertising

Media packs are available on request
Commercial Director **Clare Dove**
clare.dove@futurenet.com
Advertising Director **Richard Hemmings**
richard.hemmings@futurenet.com
01225 687615
Account Director **Andrew Tilbury**
andrew.tilbury@futurenet.com
01225 687144
Account Director **Crispin Moller**
crispin.moller@futurenet.com
01225 687335

International

Linux User & Developer is available for licensing. Contact the International department to discuss partnership opportunities
International Licensing Director **Matt Ellis**
matt.ellis@futurenet.com

Print subscriptions & back issues

Web www.myfavouritemagazines.co.uk
Email contact@myfavouritemagazines.co.uk
Tel 0344 848 2852
International +44 (0) 344 848 2852

Circulation

Head of Newstrade **Tim Mathers**

Production

Head of Production US & UK **Mark Constance**
Production Project Manager **Clare Scott**
Advertising Production Manager **Joanne Crosby**
Digital Editions Controller **Jason Hudson**
Production Manager **Nola Cokely**

Management

Managing Director **Aaron Asadi**
Editorial Director **Paul Newman**
Art & Design Director **Ross Andrews**
Head of Art & Design **Rodney Dye**
Commercial Finance Director **Dan Jotcham**

Printed by

Wyndham Peterborough, Storey's Bar Road,
Peterborough, Cambridgeshire, PE1 5YS

Distributed by

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

We are committed to only using magazine paper which is derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards. The manufacturing paper mill holds full FSC (Forest Stewardship Council) certification and accreditation

Disclaimer

All contents © 2017 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend, adapt all submissions.

ISSN 2041-3270



Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

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

Chief executive Zillah Byng-Thorne
Non-executive chairman Peter Allen
Chief financial officer Penny Ladkin-Brand

Tel +44 (0)1225 442 244

Welcome

to issue 186 of Linux User & Developer

In this issue

- » Ultimate Rescue & Repair Kit, p18
- » The Future of Programming, p60
- » Best Arch-based distros, p81



Welcome to the UK and North America's favourite Linux and FOSS magazine.

I'm writing this on the day that it was discovered that anyone can 'hack' Apple's macOS High Sierra by clicking a prompt and typing 'root' in the username field. I laughed, but I shouldn't as it's quite a ridiculous security failure that should never happen. Back in the world of Linux, we're looking at ways to reduce the chances of failures of a different kind with our Ultimate Rescue & Repair feature (see p18). We explore what to do

when disaster strikes: how to analyse issues, recover from them, and clean and maintain systems to avoid problems in the future.

For our second feature, we consider what programming languages we'll be using in ten years' time (p60). As in the world of programming, the past offers clues to what technologies will be big in the future and that's reflected in a new three-part tutorial series, where we cover MQTT. This 18-year-old protocol is extremely hot now because of the growth in IoT. We're also coming to the end of our Learn Java series (p54), so email us with what languages you'd like to see in the magazine. Thanks for joining us this year as it draws to close. We hope you've enjoyed the magazine as much as we have enjoyed making it for you!

Chris Thorne, Editor

Get in touch with the team:
linuxuser@futurenet.com



Facebook:

facebook.com/LinuxUserUK



Twitter:

[@linuxusermag](https://twitter.com/linuxusermag)



filesilohelp@futurenet.com

For the best subscription deal head to:

myfavouritemagazines.co.uk/sublud

Save up to 20% on print subs! See page 30 for details

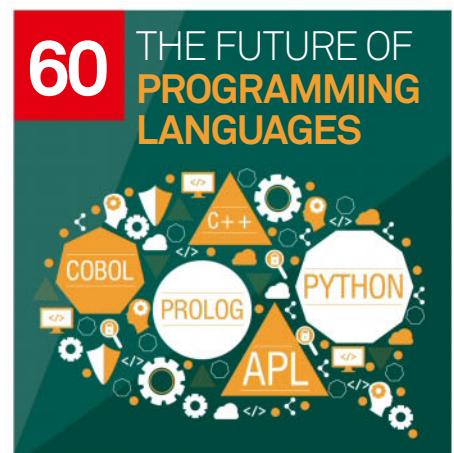
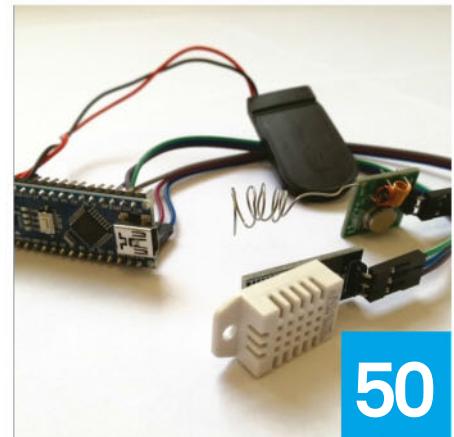
Contents



18

ULTIMATE
**RESCUE &
REPAIR KIT**

The cover features a red background with white and green medical-themed icons (gears with crosses, a first aid kit) and the title 'ULTIMATE RESCUE & REPAIR KIT' in large white letters.



60

THE FUTURE OF
**PROGRAMMING
LANGUAGES**

The cover features a dark green background with various programming language names (COBOL, PROLOG, PYTHON, API) and icons (gears, code snippets) arranged in a circular pattern.

OpenSource

06 News

Intel and AMD develop an revolutionary new CPU for 2018

10 Letters

Scribbled musings from readers

12 Interview

Vivaldi CEO, Jon von Tetzchner discusses privacy and net neutrality

16 Kernel Column

The latest news on the Linux kernel

Features

18 Ultimate Rescue & Repair Kit

While we all know that Linux is one of the most stable and reliable operating systems around, things can occasionally go wrong. However, there's no need to despair if they do – **Neil Bothwick** reveals the best tools for diagnosing and fixing system problems

60 Future of Programming Languages

Looking into his crystal ball, and extensive knowledge base, **Mike Bedford** examines the future of programming languages. Starting with unusual ones that broke the mould, he moves on to up-and-coming languages before getting the views of experts about what the future holds

Tutorials

32 Essential Linux

Explore the Joy of Sed

36 Essential admin commands

20 terminal commands you need to know

40 Video capture

Record with SimpleScreenRecorder

42 MQTT

An intro to the protocol powering IoT

46 Security

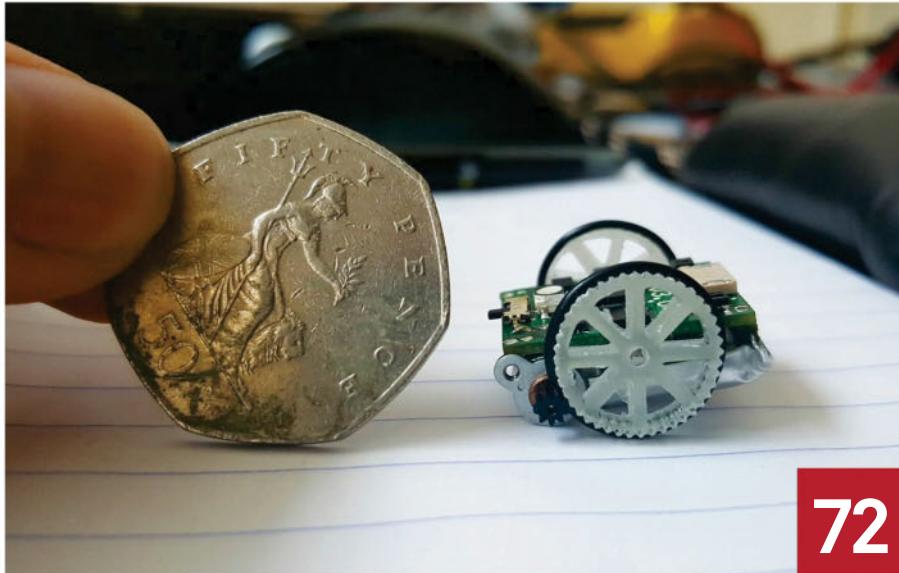
Redirect HTTP(S) requests from devices

50 Arduino

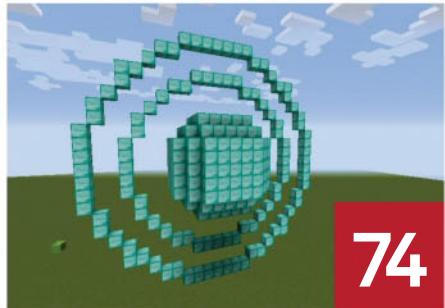
Use sleep mode to monitor temperature

54 Java

Spring Framework: dependency injection



72



74



76



86



88

Practical Pi

72 Pi Project: Micro robots!

Seeking a better way to teach multi-robot behaviour in class has led **Joshua Elsdon** to create micro robots that are smaller than the change in your pocket

74 Minecraft & Python

Learn how to draw vector shapes directly into the Minecraft world using just a few lines of Python code

76 AI assistant

Add voice-activated artificial intelligence to your Raspberry Pi with the aid of Google Assistant and ReSpeaker pHAT

78 Pythonista's Razor

Discover how to use SQLite with Python to create a database to manage all your precious information

Reviews

81 Group test

Experience the benefits of Arch in the comforts of a desktop distribution. Which one of the four solutions will come out on top?

86 Hardware

The iStorage diskAshur Pro2 external hard drive offers solid security, but is it worth the premium price tag?

88 Distro spotlight

Linux hardware vendor System76 has put together its own distro, based on Ubuntu, called Pop!_OS

90 Fresh FOSS

CudaText 1.23.0 code editor, Subsurface 4.7.1 dive log tool, LXQt 0.12.0 DE and Converseen 0.9.6.2 image converter

Back page

96 Near-future fiction

Chained to a super-smart mirror, there is a strong desire to break free



SUBSCRIBE TODAY

Save up to 20% when you subscribe! Turn to page 30 for more information

OpenSource

06 News & Opinion | 10 Letters | 12 Interview | 16 Kernel Column

HARDWARE

Intel & AMD develop revolutionary CPU

Surprise joint venture yields results: new chip for 2018 release offers high-performance, discrete graphics



What would you say to a powerful new CPU from Intel with built-in graphics from AMD? Nice idea—and something you're going to be able to buy soon. Rumours of an Intel-AMD collaboration have been swirling since early 2017, but it seemed just too far-fetched. After all, the former partners have a history of litigation between them.

Even more surprising is that the venture has yielded results so swiftly. Clearly noting the obvious weaknesses in its own graphics processors, Intel has done the sensible thing and brought on board the only company that could appreciate the engineering difficulties of balancing a CPU and discrete GPU.

But not only does this cessation of a 30-year rivalry deliver a new product; it could also revolutionise computing over the coming years, offering OEMs the freedom to develop lightweight, thinner designs. It doesn't end there: improved thermal dissipation, new cooling solutions and increased battery life are all possibilities.

"Our collaboration with Intel expands the installed base for AMD Radeon GPUs and brings to market a differentiated solution for high-performance graphics," said Scott Herkelman, vice president and general manager of AMD Radeon Technologies Group. "Together, we are offering gamers and content creators the opportunity to have a thinner and lighter PC capable of delivering discrete performance-tier

graphics experiences in AAA games and content creation applications. This new semi-custom GPU puts the performance and capabilities of Radeon graphics into the hands of an expanded set of enthusiasts who want the best visual experience possible."

Introduced as part of the 8th Gen Intel Core series, the project unifies Intel's high-performing Intel Core H-series processor, its second generation GDDR5 High Bandwidth Memory (HBM2) and a custom discrete graphics chip from AMD's Radeon Technologies Group. Intel's Embedded Multi-Die Interconnect Bridge (EMIB) technology is at the heart of this, supporting the processors with a revised power-sharing framework. Basically, EMIB is a bridge that speeds up the flow of data, enabling faster, more powerful and efficient devices.

"Intel has brought on board the only company that could appreciate the engineering difficulties of balancing a CPU and discrete GPU"

But what does all this mean to GNU/Linux and the open-source world? At this stage it is too early to say, but thinner devices aren't just benefits for gaming. Mobile hardware, media centres and smart home devices are



Above Intel Inside or should it be AMD? If it's thin and lightweight portable it can now be both!

all obvious homes for this new technology. Meanwhile, notebooks, hybrids, all-in-ones and mini desktops are all set to become lighter and more powerful.

As for gaming, thinner devices, integrated gaming TVs and lighter games tablets are all possibilities. The implication for home-based media production, meanwhile, is considerable. But some things are less clear.

So far, we don't know how much the new processor will cost. Similarly, neither Intel nor AMD has advised whether the processor will be available beyond the OEM market.

Oh, and it doesn't even have a proper name yet. Intel Core Single Package Multi-Chip Solution doesn't exactly roll off the tongue, does it?



WEB DEVELOPMENT

Firefox 57 “Quantum” challenges Chrome

Faster and stronger browser hits Ubuntu

Billed as the biggest update to hit Mozilla Firefox in the browser's 13 years, Firefox 57—dubbed “Firefox Quantum”—is here. But does it make the quantum leap needed to combat the popular browser's decline?

Featuring 4,888,199 new lines of code and taking a year to produce, Firefox Quantum has taken an important step into the future. This version of the browser finally casts off the shackles of the legacy extensions, providing compatibility only with web extensions. While this means a good number of older add-ons will no longer work, 6,000 working web extensions are listed. A much-needed stability and security improvement, the benefits outweigh the shortcomings.

“We couldn't do any of this without our loyal and rabid users,” blogged the Firefox Quantum team, “Who make us try harder, work later, code longer, cheer louder and fight for more and more of what's right in keeping the internet open, safe and exciting.”

So can Firefox fight back against Chrome? Although it will take a while for the figures to reflect it, the early signs are good. Most of the excitement is focused on the

performance improvements, which are widely considered to be twice as fast as 2016's Firefox 49. Notably, Quantum uses 30 per cent less memory than Google Chrome, partly due to changes under the hood, and also thanks to the new CSS engine, Stylo, and a new multi-process architecture. Public reaction on Twitter and Reddit has been positive, and there's a buzz around Firefox that hasn't been seen since the noughties.

Firefox Quantum offers a number of features to enhance the browsing experience. Rather than system screenshots, Firefox enables you to snap them in the browser. Meanwhile, the Pocket read-later-syncing service is built into the browser. If you have Firefox installed on other devices, you can keep reading. Pocket saves, screenshots and even tabs can be found in Firefox's own cloud library. The browser also supports WASM and WebVR for next-generation gaming, new themes and toolbar reconfiguration.

Firefox Quantum can be manually downloaded from www.mozilla.org, although Ubuntu 14.04 and later users will receive the update automatically.

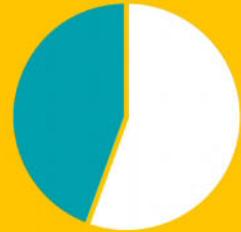
DISTRO FEED

► Top 10

(Average hits per day, month to 17 November 2017)

1. Mint	▲ 2766
2. Ubuntu	▲ 2032
3. Debian	▼ 1825
4. Manjaro	▲ 1616
5. Antergos	▼ 1323
6. Solus	▲ 1321
7. elementary	▲ 1090
8. Fedora	■ 960
9. openSUSE	■ 930
10. TrueOS	▼ 857

► This month



Another Debian-heavy month in the top 10. Meanwhile, several distros are gaining interest for their Windows-like desktop environments...

► Highlights



PCLinuxOS

Perhaps the most famous of the Windows-like, PCLinuxOS comes with out-of-the-box support for many graphic and sound cards, and runs most popular hardware. The grey toolbar and menu deliver a strong Windows feel, while a user forum exists for troubleshooting.



Zorin OS

Based in Ireland, Zorin OS is perhaps the ultimate Windows-esque Linux distro. As well as offering a start menu experience similar to Windows 10, Zorin OS ships with Wine pre-installed.



antiX

Fast and lightweight, antiX uses a classic Windows-style desktop toolbar and start menu reminiscent of Windows XP. Based on Debian, antiX can also run on old Pentium III systems.



Latest distros available:

filesilo.co.uk

DISTRO

Slax abandons Slackware for Debian

New release retains Slax name, not yet being renamed “Dex”

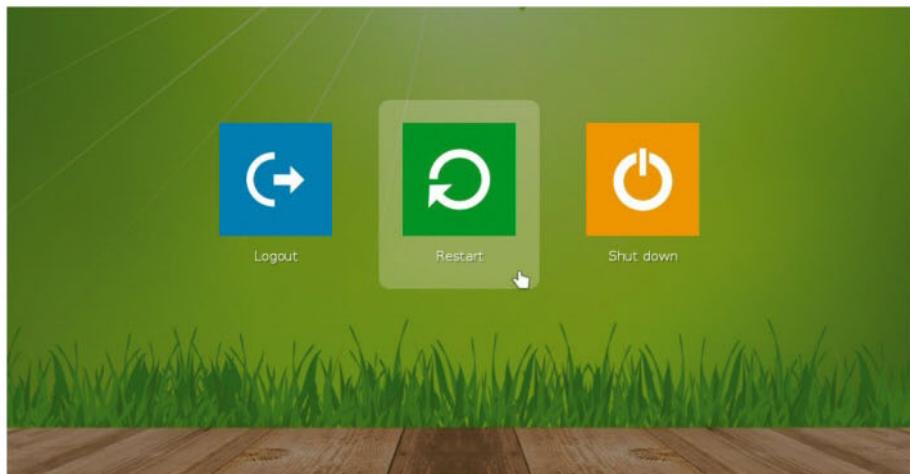
Slax 9 is now available for download, but in a surprise move developer Tomas Matejicek has announced that the new version of the lightweight distro has abandoned its roots.

So if Slax isn't based on Slackware, what's running underneath the leafy new desktop background? Well, it's Debian. Outlining his reasons with clarity, Matejicek has basically stated that it's too complicated to maintain Slax if it continues to be based on Slackware.

“The reason is simple: laziness,” Matejicek blogged. “I am too lazy, really, really lazy. When I prepared Slax from Slackware, I had to patch kernel with aufs, configure, recompile, and so on. Then compile other software from sources, fight dependencies, and so on. I enjoyed doing that in the past, but now I'm not in the mood anymore.”

To prove Matejicek's key point (“all Linux distros are the same anyway, ... it's all Linux”), the new version of Slax

“What's running underneath the leafy new desktop background? Well, it's Debian”



Above Slax 9 looks great, but has abandoning Slackware sacrificed users?

is just 208MB (or 218MB for the 32-bit release). Fitting comfortably on all but the smallest USB sticks, there's probably just one key argument against Slax becoming (unofficially) Dex: the arrival of systemd.

Meanwhile, the desktop arrives with just four icons: the Terminal, Text Editor, Calculator, and Chromium Browser. Matejicek's rationale is simple: he believes

everything is moving to the web, so why spend time packaging tools into an OS?

Slackware has remained systemd-free, with the bootstrap and process manager often cited as a tipping point for Slackware adherents. It seems likely the move to Debian (and implicit acceptance of systemd) will lose some users, but Matejicek seems confident that it doesn't really matter.

PUBLIC SECTOR

Pentagon switching to open source in 2018?

DoD says updates to closed-source software are too slow

Could the Pentagon be abandoning proprietary software for open source? That's the implication of a key portion of the National Defense Authorization Act for Fiscal Year 2018, in an amendment introduced by Sen. Mike Rounds and co-sponsored by Sen. Elizabeth Warren.

But why? First, as the world's biggest single employer, the Department of Defence has a massive IT requirement. A lot of money is spent on data warehousing, statistics, briefings, presentations, spreadsheets and documents. Licencing costs are expensive.

Second, there is a perception that waiting for closed-source applications to update puts the DoD at a disadvantage. Rather than playing catch-up, it's preferable to adopt new software tools. Procurement takes time, both in terms of signing off requests and waiting for a release date. Then there's the roll-out, another potential delay.

Of course, this amendment is not without its detractors. The security argument is being highlighted. For instance, Brian Darling, president of Liberty Government Affairs, has written in the conservative publication

Townhall challenging the proposal, citing national security concerns. Conversely, Federal Chief Information Officer Tony Scott called in 2016 for government-specific open-source code to be built, demanding it be “secure, reliable and effective.”

Supporters of closed-source solutions seem to have overlooked that international rivals have personnel capable of spotting and exploiting flaws in proprietary software. At this stage, and in the wake of Munich ditching Linux, it seems likely the amendment will be defeated.



HARDWARE

Critical bugs found in Intel Management Engine

Security vulnerabilities also identified in SPS and TXE technologies

Using an Intel PC or laptop? Your computer is at risk. Hidden away in Intel's Management Engine (ME)—the secret 'computer within a computer' whose existence was only confirmed in 2016—are a collection of bugs that could enable any Intel-equipped Linux, Apple Mac and Windows computer to be hijacked, even with no operating system installed.

Third party security experts (UK-based Positive Technologies) have discovered that along with ME, Server Platform Services (SPS) and Trusted Execution Engine (TXE) are susceptible to exploitation by anyone posing as a network administrator. The attacker may then install spyware and rootkits. Once a user is logged in, malware or hijacked applications can be subverted to leak data from the system RAM. Shared machines and servers are particularly at risk, and code to exploit ME can be introduced via USB stick.

As Intel's Management Engine was first introduced in 2008, the overwhelming majority of processors currently in use have the vulnerability. This includes 6th, 7th, and 8th generation Intel Core CPUs, low-end Celerons and Intel Xeon server chips.

Described as a 'backdoor' by some security commentators, ME is essentially a mini computer with its own CPU and operating system (since 2015, the UNIX-like MINIX). Perhaps most worryingly, ME runs beyond your OS, out of the reach of antivirus tools.

So why is it included? Intel's Management Engine is ostensibly provided to allow network admins to remotely access servers and workstations, fix errors, provide desktop support and reinstall the OS from afar, which are all vital tools in many organisations.

Positive Technologies' researcher Maxim Goryachy explains: "An attacker could use this privileged access to evade detection by traditional protection tools, such as antivirus software. Our close partnership with Intel was aimed at responsible disclosure, as part of which Intel has taken preventive measures, such as creating a detection tool to identify affected systems."

Intel's detection tool (INTEL-SA-00086) can be grabbed from its download center. Meanwhile, patches have been rolled out. However, don't expect any patches from your operating system. If your system is affected, you'll need to contact your hardware's OEM.

TOP FIVE

Open source web browsers



1 Mozilla Firefox Quantum

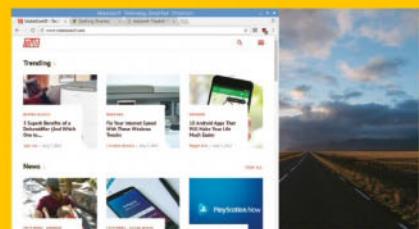
Released amid a storm of publicity, Firefox 57 (known as Quantum) has proven popular. The biggest update in Firefox's history, Quantum is faster than Google Chrome, and uses 30 per cent less memory. These changes may secure Firefox's future, but are they enough to challenge Chrome?

2 Midori

Employing the WebKit rendering engine, Midori is associated with the Xfce desktop and developed to the same principle ("making the most of available resources"). With a low footprint, it can be found on many light Linux distros.

3 Chromium

Anyone who has used Google Chrome should be comfortable with its open source sibling. Much of the same functionality is there, along with support for extensions. Chromium is fast and easy to use.



4 Brave

Based on Chromium and spearheaded by Mozilla Project co-founder Brendan Eich, Brave shares less data with advertisers and blocking trackers.

5 GNOME Web

Formerly known as Epiphany, this is still going strong (the package retains the name epiphany-browser), and utilises the WebKit rendering engine. Recently, some integration with Mozilla Firefox Sync has been added, enabling continuity across devices.

COMMENT

Your letters

Questions and opinions about the mag, Linux and open source



Above What events would you like to see covered next year? Email us at the address below

Key conferences

Dear **LU&D**, I started reading your Linux magazine about a year ago. It's a great read, and I have learnt a vast amount from the tutorials and the more practical features that you publish. I have also enjoyed your pieces on general Linux goings-on, stuff like Linux in Space. I have to say that one thing that has surprised me is the lack of an events page. I could do with something that tells me what training events and conferences are worth my time. I think that would be helpful to busy readers like myself who only want the key diary dates.

Joshua Allen-Phelps

Chris: Glad you're getting so much out of the magazine, Joshua. Thanks for the topic suggestions that you supplied in my reply as well, it's really helpful to get an idea of what people want to see in the magazine. Diary dates would be a good idea, but I'd be interested to see whether more people want this and what they'd like to see. Should we supply a Linux User Group (LUG) list, stick to professional events or do both? My concern would be to make sure we only cover the crucial events, as your time out of the office is valuable. We'd also love to hear from organisers and attendees, as we're always on the lookout for people who can report on the events for us. Email as at the usual place: linuxuser@futurenet.com.

GET IN TOUCH!

Got something to tell us or a burning question you need answered? Email us on linuxuser@futurenet.com

```
root@rooter-desktop: ~ $ lsmod
Module           Size  Used by
nls_utf8          16384  1
i3c              40960  1
nvram           16384  0
video            40960  0
msr              16384  0
nvidia_uvm       651264  0
nvidia_drm       49344  1
nvidia_modeset    790528  5 nvidia_drm
nvidia          12312576  91 nvidia_modeset, nvidia_uvm
dm_crypt         28672  1
kvm_vmx          73728  0
input_leds        13280  0
jbd2             20480  0
kvm              598016  1 kvm_amd
irqbypass        16384  1 kvm
snd_hda_codec_hdmi 458048  6
k3b              15376  0
serio_raw        16384  0
drm_kms_helper  167936  1 nvidia_drm
drm             366040  4 nvidia_drm, drm_kms_helper
snd_hda_codec_via 24576  1
snd_hda_codec_generic 102400  1 snd_hda_codec_via
fb_sys_fops       16384  1 drm_kms_helper
syscavorearea    16384  1 drm_kms_helper
snd_hda_intel     36864  5
sysfirmware      16384  1 drm_kms_helper
udf_scsi_gdi     16384  0
```

Right Making sense of the output of useful commands such as `dmesg`, `fsck` and `lsmod` will assist those trying to help you

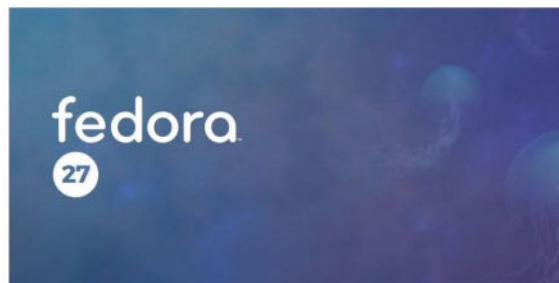
Crucial commands

I've got a suggestion for a subject to cover in your excellent Linux magazine. I find that when things go wrong and I head for the usual places to seek advice from more experienced heads that a post without some useful output can either get dismissed out of hand, or my post is swiftly followed by a request that I run various sysadmin commands.

Knowing about the best ways to read the outputs of these commands is an important skill as well. Also, the ways that you can use commands to decipher the output would be appreciated too. There are so many options to choose from in the main pages, it's sometimes hard to know what's best. I know you've covered grep in the past, but there are other methods.

Derek Harris

Chris: Thanks, Derek. Some of those commands are likely covered in our Ultimate Rescue and Repair feature on p18 of this issue, but what a great idea! It's certainly something we can look at doing, especially as John Gower's little series on mastering shell scripting is drawing to a close. I'm sure forum veterans would appreciate posts for assistance that demonstrate some thought into what information might help in getting to the root of the problem. Watch this space and we'll see what we can do.



Above Fedora 27 won a recent poll to be included on the LU&D disc, but what distros would you like to see in the future?

Ubuntu out

Dear **LU&D**, isn't it time to drop all the Ubuntu coverage and launch packs? There's so much more going on in the desktop world beyond Canonical, especially after it has dumped the desktop in favour of lots of dough from cloud and IoT customers.

Larry Smith

Chris: I wouldn't say Ubuntu has "dumped the desktop"; it still has a 17-person team working on the desktop version of Ubuntu. Admittedly, most of those are GNOME developers, but that's still a sizable team. In some ways, Ubuntu's focus makes it more relevant to **LU&D**. From the dribble of feedback we're beginning to receive, it's clear that the 'Developer'

portion of our name needs more love. We've tended to focus almost entirely on the desktop OS story, but that needs to change, as the budding sysadmins and pros also need to know what features are going to benefit their work-life. You're right that there's a lot of exciting things happening in the desktop universe—that's why we have Antergos on the disc—but we always want to hear what distros have caught our readers' attention. We'll likely cover Solus soon, but we also ran a poll to see if readers wanted Fedora 27. It gained 65 per cent approval from our Twitter followers (out of 285 votes). Not exactly a ringing endorsement, but enough for us to put it on the disc.

Pixel OS in

Dear **LU&D**, I'm interested in the Raspberry Pi, but I'm mostly keen on trying out the Raspbian OS. Can I install that on one of my old laptops as well, or do I need to buy one of the RaspPi boards?

Philip Booker

Chris: Yes you can, Philip. In December of last year, the Raspberry Pi Foundation released Pixel OS, an x86 version of Raspbian that uses the PIXEL desktop environment (which is modified from LXDE). The OS is particularly suited to ancient machines and if your interest is based on coding, you'll find many environments bundled with it (including Python and Java) as well as environments for Pi add-ons such as the Sense HAT emulator. You can download it here: <https://www.raspberrypi.org/downloads/raspberry-pi-desktop>.

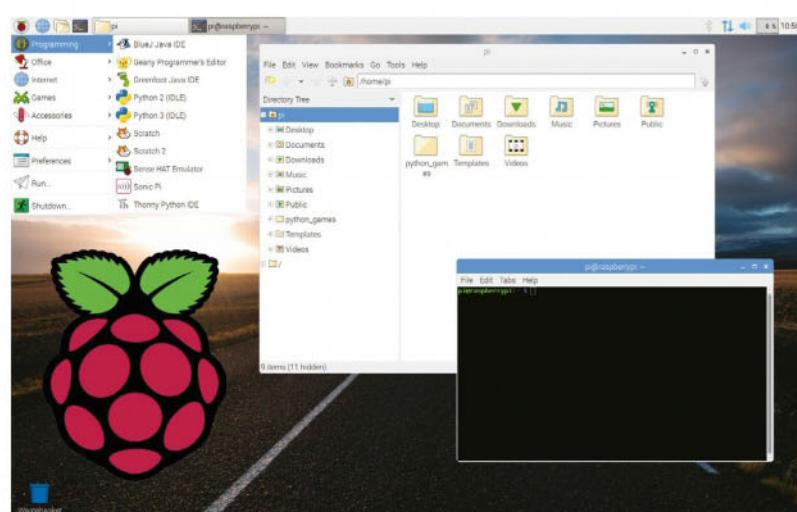
TOP TWEET

@andysc:
"Yes, it's been a long journey, but very pleasing to see the "dream" come true :)"

Andy Stanford-Clark, the co-author of MQTT, answering what it feels like to see the protocol taking flight 18 years later in today's world of IoT devices.

Follow us on Twitter
@LinuxUserMag

Below Pixel OS will have its first anniversary as an x86 desktop OS in December



INTERVIEW VIVALDI

Vivaldi: The power user's browser

We interview the CEO of Vivaldi, a Chromium-based browser, about the fight for privacy and net neutrality and what features are in the pipeline

**Jon Stephenson von Tetzchner**

has been blazing a trail in the internet-browsing business since 1995 as co-founder of Opera, and now as the CEO of Vivaldi.



The Vivaldi browser started life in 2015 as both a response to a change of direction at Opera and what Jon Stephenson von Tetzchner, the CEO and past co-founder of Opera, sees as a demand for a web browser that caters for internet power users.

Von Tetzchner believes that the main browsers are over-simplifying their experience: "They are basically adapting to someone that doesn't exist," says Jon talking from Magnolia Innovation House outside Boston. "Any feature that's used by not so many people gets removed and the software gets dumber and dumber in a way." Vivaldi is essentially walking in the opposite direction with signature features such as tab grouping, a built-in screenshot capture tool and the ability to take notes right in the browser that are heaped together with lots of customisation options. And while many browsers rely on add-ons, Vivaldi is releasing differentiating features that are built natively into the browser.

Can you tell us about your overarching philosophy for the browser?

Our feeling is that if you're spending a lot of time on the internet, you actually need a browser that can adapt to your requirements and your needs. There should be multiple ways to do things. In some ways, in the way that you do things on Linux, you have a lot

of flexibility, a lot of different ways to do things, so you find your way to do things and that's what we're all about. It's about adapting to your requirements as a user by putting in a lot of features, but by also putting in a lot of ways to do the same thing so you can have it adapt to your requirements instead of the other way around.

Have you experienced a notable interest from Linux users?

Yes. If you look at the amount of Linux users we have, they match the number of Mac users, which tells you that Linux users are obviously flocking to us in much greater numbers than users of other platforms, which is nice.

Are you around the 1 million users mark?

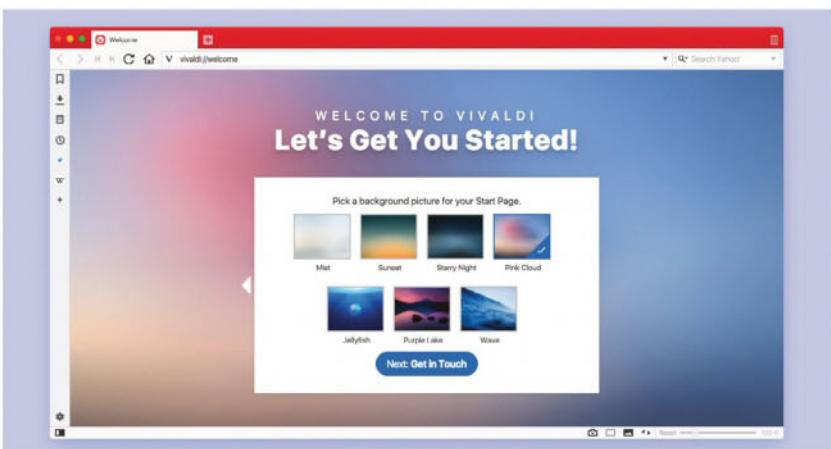
It's around that figure, we are saying that we're closing in on 1 million. That's kind of where we are. We're getting pretty close to that. Our growth rate is increasing, we're close to our first million and we're counting active users; we're not counting downloads [...] or numbers that make no sense. We're counting what means something to us, which is the number of active, monthly users.

The Chrome browser is dominant on the Linux platform, even Mozilla has acknowledged that, and the amount of data being tracked now is significant. Do you see Vivaldi as a solution?

We will do what we can on our side, but actually, after spending a lot of time thinking about tracking and targeting, my feeling is actually this is a problem that needs to be solved in a different layer. That's something I've decided to start talking about. I was getting the questions, "What are you doing to help keep us safe?" and we started discussing whether we should put in Tor or something like that. There are a number of things we can do, and we're looking into that. The trouble is that the moment you start using something like Tor, you actually look like you have something to hide. And that's how the government views it, and I think that's unfortunate that we, when we want to just be private, we're seen as wanting to do something wrong. My feeling is that this needs to change, and there is a problem with the amount

Below Vivaldi is striking out in the opposite direction to the main web browsers by offering multiple ways to do things

VIVALDI



What's new and coming?

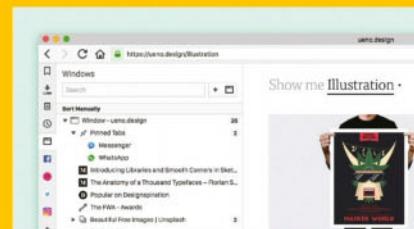
What's impressive about Vivaldi is the features that are being released with each update. For instance, the last release, 1.13, focused on the Window Panel. This is a classic example of Vivaldi catering for power users who tend to have lots of tabs open. The panel presents a 'tree-style' view of tabs to the side of the browser window for easy management.

You can do such things as grab and drag tabs; group them by topic; use Tile Tab Stack feature to compare pages side by side, mute individual tabs or hibernate tabs for better performance: "it's another way to do tabs," says von Tetzchner, "and we just know that people have very different opinions and strong opinions how they should do their tabs."

The Vivaldi team has worked on improving the Downloads panel as well, as von Tetzchner says: "We view downloads

differently from the other browsers. We have the download panel [...], which has a lot more information." This time, Vivaldi has included the ability to pause and resume downloads; it's also added supply speed information in the progress bar, and a warning dialog when closing the browser mid-download, which are all important for Vivaldi users according to Jon, as they download more than the average user. He also says that while Vivaldi doesn't have BitTorrent currently, it's one of the items on the development list, as it was included in Opera.

In terms of the future, while von Tetzchner says a lot of the focus is on finishing off existing tasks, such as the email client and mobile, he's got his eye on peer-to-peer. "From a privacy perspective, I think that's great. I don't think everything has to go through the



Above Vivaldi is constantly adding features. Recently that's meant a new Window Panel for management tabs (pictured), but synchronisation is due late November

cloud [...]. Too much is going through the cloud, which is a security and privacy issue," says Jon.

In the short-term, Vivaldi is due to release its most most-requested feature: synchronisation on 29 November, and in early December it will release an experimental build for the Raspberry Pi and ARM devices.

of information that is being collected, as it is too great and the amount of targeting is too great. We're somewhere on the way, just because the technology provided us with the way to do this, the industry said that's okay to do, and I think it's wrong. The problem needs to be fixed at a different level, basically.

You have talked about regulation in the past. What form of regulation do you feel is required for online search and advertising?

There are two things, and in some ways it's not that big a change. It's basically going back to where we were a few years ago, and I'm not the only one who has been talking out about this—Sir Tim Berners-Lee has also made comments in this direction. It's something that has been seen by a number of us who have been on the internet for a long time that something is going in the wrong direction here. It's rather simple: you shouldn't be allowed to collect more information than you need, and you shouldn't be able to use the data for things that are not relevant to the task that's being performed.

As an example of that, there is no need to collect information about your location at any one time, and there's no need to provide that to third parties either. I mean, the location information is needed when you as a user want the location information. That's when that information is useful. So you want to use it for maps, you want to use it for driving in traffic, or maybe want to have it on for a while because it helps you. There are two things: how much are you

collecting and keeping it, and how much are you selling it? Or using it for other purposes that were intended? There are a lot of things I can say I don't mind—for example, the concept of recognising voices is really nice, right? And being able to translate, the Google stuff that's shown is fantastic, but the question is, can we trust Google not to use that information for anything else? Does it need to go into the cloud at all for the translation purposes? This is the question and what we've been seeing, at least with the collection of data and how it's being used; it's being used in forms that have gone too far.

It used to be that advertisements were based on the location. You'd know something about the audience that was in that location and you would be able to use IP to decide where in the world they are from, but you weren't able to do localised advertisements based on the person. You weren't able to follow a person from page to page. That praxis is very unfortunate, and needs to be changed.

You have stated in the past that advertising data should be freely available. How do you envisage that working?

If you see an ad, whether it's coming through Facebook or Google [...] you should be able to see who provides that ad. You should also be able to see all the ads provided by that same party to get an impression of what this person is doing. I don't think that's enough to deal with how ads are being used by some. I mean, supposedly Trump sent

“It's unfortunate that we, when we want to just be private, we're seen as wanting to do something wrong”

The licensing elephant

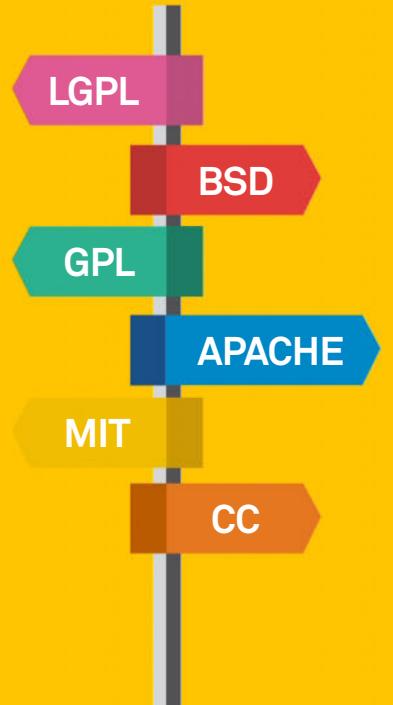
You can't cover Vivaldi without addressing the elephant in the room, and that's the browser's current lack of a full free or open source licence. It is a complicated situation, as all the changes made to the Chromium engine source code (available at <https://vivaldi.com/source>) are released under a BSD licence. This type of licensing doesn't require source code to be distributed, although Vivaldi does, and also makes the UI code accessible to read in plain text.

"Every change we do to the C++ code—that's open," says von Tetzchner. "When it comes to the HTML side, we haven't decided on a licence. We have had discussions—should we put it out with a licence? But we've not really done that." And people are able to go in and change the code as long as they know HTML, JavaScript and CSS, but Vivaldi isn't actively encouraging it: "People can't then take that and build their own [browser] on that, but at the very least people are able to go check the code and have a closer look at what we're doing—that we do like."

Von Tetzchner says that the company did consider using GPL, but that it's a difficult

decision because of the risk involved: "By picking the wrong licence and putting it out there, we're really concerned by that, and we do feel that, in many ways, people can still do what they like," says Jon. It also hasn't deterred the community from contributing code, as some users have put out patches and sometimes submitted patches for incorporation. Sometimes these patches have been included in the browser, but von Tetzchner acknowledges that mostly they don't make it in.

It seems that Vivaldi's biggest fear is the distribution and sharing requirements of some free and open source licenses: "That's probably our biggest risk, yes," says von Tetzchner. "Our impression is that people like what we're doing [...] I think the way we do things as a company, we're more open than anyone else [...]. It's all very open with how we work with people. The source is not something that comes up on a regular basis, it's mostly when I talk to Linux magazines, which is reasonable, and it is a difficult thing, as a lot of us like Linux and a lot of us like open source, so it's complicated."



Above Vivaldi's CEO says that the company faces a "difficult decision" when it comes to what unified open source licence to use for its innovative browser

out 40,000 different ads per day [During the US Presidential Elections] and clearly going through that is not feasible, so I would say that it needs a programmable interface to that where you could use big data to actually analyse the ads that are being sent out [...] so we can analyse if someone is doing the kinds of ads you don't want to be seeing; that people are sending out different ads to different people with a different message to get them to act in a certain way. I mean, from a democratic perspective that's important for us to know when people are doing that.

Our online services have not been regulated in the same way as maybe television and other channels have been, and we are seeing this is an even more potent channel that needs to be, if anything, more closely scrutinised. We've seen some companies given tools to do bad things will do bad things. This is a very powerful tool to do bad things if you want to.

How does Vivaldi make its money if it's not tracking user data?

Most of the revenue from browsers is from search, so we have a few search deals, and then we have a few bookmarks that we include that generate revenue [...]. The amount of revenue we're calculating

with is about a dollar per user per year. We can do things without spying on our users, and I think it's a lot more fun building great products for our users and being on their side, and okay, if we can make a dollar per user per year, that's enough to pay our bills.

Net neutrality is in serious danger in the US. Do you think there's a way round the repeal?

You always have to be optimistic. I fear that it will be a negative decision this time. We have to hope that it will be repealed if that happens, and the people doing that will pay for taking the side of big corporations against the public.

I think it's good that people are speaking out, we've been doing that ourselves as well. In the last round, we did that forcefully. In a way, it's something that we all need to speak out about then follow up closely with the companies. Potentially switching companies if we are finding our companies are doing bad things.

This is the second very important change that is happening. The first one was the decision to let telcos collect information to the same degree as Facebook and Google, providing, in some ways, more parity between them. That's the argument,

which obviously we don't buy. I think it should be the other way around: we should restrict Facebook and Google in their collection of information, not giving more people and companies the chance to spy on us.

What's your community approach?

It's the reason why we're doing Vivaldi in the first place. Before we were doing Opera, we had a very close relationship with our community, and when Opera strayed from the straight and narrow and started doing something totally different, a lot of people were unhappy about that, and we felt we owed them not to leave them stranded. So we started to build a browser, and we're doing that in very close relationship with our community.

We have very different levels; we have volunteers helping us test and give us feedback. We have volunteers helping us to translate. We then have a very active community on <https://vivaldi.net>, and users are very vocal about what they think we should be doing with the software, what we should be doing better and what we should improve. We take their feedback really seriously. We have this motto, 'We are building this browser for our friends'. It's not just a saying; we look at our users as our friends, and we want to build a great browser for them. When you think about it, you don't spy on your friends; you try to keep your friends safe, and you want to listen to your friends' advice—for us it's real. We want to do the best for our users, it's also one of the reasons why we're speaking out about the privacy issues, because we think it's important, and if you can't fix everything on a technical level we can engage in other ways as well.

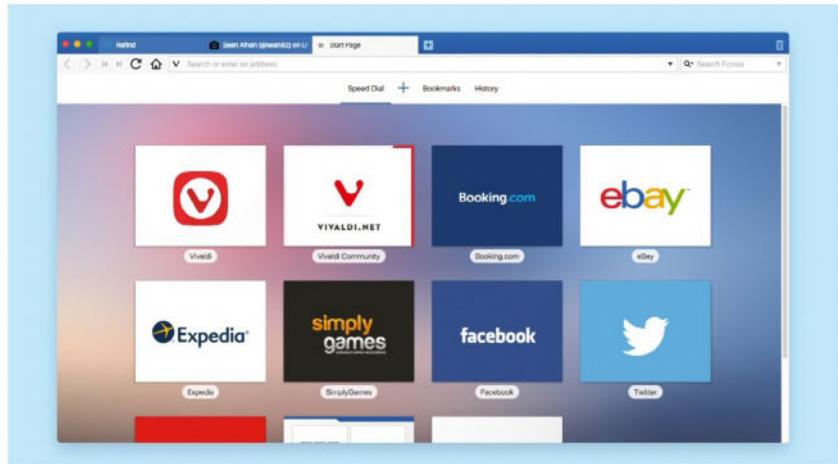
I noticed you had a blogging platform.

We're providing a blogging platform and a webmail service which has no ads, and we're not going through it, except for viruses and things like that, but it's part of us providing a platform for our users to hang out and communicate with each other, and tells us what they want and don't want.

Is the webmail separate to the mail client to what you're working on?

Yes, it's separate. We're working on the mail client as well. The mail client works with any IMAP service, so it will work with our webmail service, it will work with Gmail even though they do things a bit differently, but any of these other services [...]. We know it's an important feature for our users, and it's getting there. I mean, I've been personally using it for a long time, but it needs to get to a shape where we feel it's right to send it out to the users. There are few things we need to fix there first, but it's shaping up nicely.

You're also working on a mobile version, is that right?



Above Version 1.12 added an Image Properties feature for viewing meta data. **Top** Vivaldi has bookmarks or 'Speed Dials' on the Start Page for accessing your favourite sites

When we started, we had mobile as well, and we ran into some technical issues, but we're working on it. It's making progress, but it's still not ready yet. I'm using a very early version myself, but it's not ready for prime time.

What is the most requested feature so far from the community?

Mail is very high, but I would say that sync is the highest. We haven't had synchronisation in yet, and that's coming very soon. [After the interview, we were advised by Vivaldi that sync would be coming in the next snapshot on 29 November, so it will be available by the time you read this.] That's also something that we've been using internally for quite some time. Part of this is that synchronisation is not only a client side thing, it's a server side thing as well, so building server technology for that is something that we've been doing. It should be ready soon, but we always say 'when it's ready'.

At the time of going to press, Vivaldi was planning to release an experimental build for ARM devices – including the Raspberry Pi, CubieBoard and ASUS Tinker Board – on the 5 December. ■

“We want the best for our users, it's also one of the reasons why we're speaking out about the privacy issues **”**

OPINION

The kernel column

Jon Masters reports on the latest happenings in the Linux kernel community, as development on new features for 4.15 closes with the release of 4.15-rc1



Jon Masters

is a Linux-kernel hacker who has been working on Linux for more than 22 years, since he first attended university at the age of 13. Jon lives in Cambridge, Massachusetts, and works for a large enterprise Linux vendor, where he is driving the creation of standards for energy-efficient ARM-powered servers.

Linus Torvalds announced the release of Linux kernel 4.15-rc1, following the “usual two weeks of merge window” (the period of time during which disruptive changes are allowed into the kernel tree). In his mail, Linus notes that this is “about the only thing usual about this merge window”. Due to the US Thanksgiving holiday, many developers (including Linus) were on vacation at the back end, and while Linus had warned everyone to front-load their patch pull requests, he did have to get much more strict about what he was willing to take at the last minute. Overall, he was fairly happy about the process, and in particular “really liked” enforcing that patches had to have flowed through Stephen Rothwell’s linux-next tree before going to Linus.

This development cycle is expected to be a little different from those in the recent past since it lines up so well with the Christmas holidays. If things go according to plan, 4.15 final will be out in the first week or two of the new year, but we are anticipating a few bumps along the way. We’ll come back to those in a future issue, meanwhile let’s look at some of the new features that are coming to a Linux 4.15 kernel via your favourite Linux distribution.

RISC-V update

Linus pulled version 9 of the RISC-V support patches into 4.15. RISC-V is a fully open source computer architecture. An architecture describes the fundamental instructions (machine code) and assembly language that any machine compatible with that architecture is capable of executing. It also specifies certain behaviours, such as the ‘ordering’ of memory operations and how interactions take place between multiple processors on a chip. With the combination of an architecture specification, reference hardware, and some skilled experience in porting Linux, it is possible to do what Palmer Dabbelt did taking the lead role in upstreaming support for RISC-V.

RISC-V is particularly interesting because it is open source. This means that not only is the architecture specification publicly available (which is common – multi-thousand page manuals are

available online for Intel x86 and ARM, among many other possible architecture choices) but that it is developed in the open using mailing lists (just like any other open source project), and that the reference hardware designs are fully open as well. What does it mean to have an ‘open source’ chip design? Well, this means that you can download the RTL (hardware description) written in Verilog source code and actually see how every part of the implementation is realised.

Using special devices known as FPGAs (field-programmable gate arrays), you can ‘synthesize’ the RISC-V designs onto a reprogrammable FPGA capable of booting Linux. Then, it is possible to make changes to the design and run those also. For those who aren’t quite aspiring hardware hackers, there are now (relatively inexpensive) RISC-V development boards that provide all of the pieces in a similar fashion to a Raspberry Pi or other single-board computers. Many more such boards will come to market over the next few years, both at the big end (capable of running Linux) and at the much smaller IoT widget end of the spectrum (capable of running an open source IoT OS like the Linux Foundation Zypher).

The RISC-V community is just hitting its stride. They have regular gatherings where new architecture and platform features are debated and worked on, and it certainly feels novel to see a group openly developing what has been traditionally a very secretive process. It’s currently a little more involved to actually run Linux on RISC-V hardware because the initial upstream port doesn’t support much in the way of devices beyond the CPU. As Palmer says, “While what’s there builds and boots [...] it’s a bit hard to actually see anything happen because there are no device drivers yet. I maintain a staging branch that contains all the device drivers and cleanup that actually works, but those patches won’t be ready for a while”.

A question of trust

Linus pulled support for AMD’s ‘Secure Encrypted Virtualization’ into 4.15. This novel feature of recent AMD CPUs allows administrators to create virtual



machines that are fully encrypted from the point that data leaves the package of the CPU. Therefore, all memory reads and writes to DRAM chips are performed using encrypted data that cannot easily be decrypted even by very sophisticated and well-funded actors with many available tools. Since the encryption happens on-chip, an attacker would need to have combinations of tunnelling electron microscopes and the expertise to work around carefully placed physical security precautions on the chip. Thus, this renders moot almost all practical attacks against the new feature. At the same time, SEV opens up many new opportunities for virtual machine hosting companies in far-flung parts of the globe. Rather than trust that a company hosting a VM is not siphoning off your data, it is possible to have that trust lie between you and AMD as the chip vendor: the hosting company is simply operating the hardware and paying the bills, but has no access to data.

RCU work and Linus on time

Paul McKenney is a (very) well-known Linux kernel developer and original author of the Linux RCU (read-copy-update) mechanism. RCU allows for so-called 'lockless' updates of certain data structures at runtime by taking clever advantage of the ability for certain algorithms to work with (slightly) out-of-date values, so long as a careful means is provided to handle when updates become visible. RCU is heavily used in the Linux networking subsystem and in other performance critical code paths. In

“RCU allows for so-called ‘lockless’ updates of certain data structures at runtime”

addition to authoring RCU, Paul has been working on fully documenting the 'memory model' used by the Linux kernel. Memory models are complex and pretty scary things. They describe how multiple readers and writers within a shared multiprocessor environment see and operate upon shared memory. They're important because they help (on some level) to ensure that many of the fundamental assumptions programmers have about how programs should operate actually remain true.

An RFC (request for comments) patch entitled 'sched: Minimize the idle cpu selection race window' aims to close a long-standing 'race condition' with CPU switching between programs (known as

'tasks' within the kernel). On a multicore and/or multithreaded system with multiple 'sibling' threads sharing the same LLC (last level cache), it is possible that a call to `select_idle_sibling` will attempt to wake up more than one task at a time. Those tasks will then race one another, with one of them ultimately winning and the other possibly being migrated (at some cost) onto another CPU. The patch aims to reduce the race window to a few instructions. Benchmarks were inconclusive but showed a modest performance improvement.

Marc Gonzalez started a lengthy back and forth between Linus Torvalds and Russell King (the 32-bit ARM subsystem maintainer) about delay handling

in the Linux kernel. Marc writes device drivers for embedded systems that are designed within his company (apparently by a guy who sits diagonally across). Like all device drivers, there are periodic requirements for

small delays between writing values into hardware 'registers' while the devices process the data and take some action. Linux has a rich framework for handling time, but Marc's annoyance was that some of these primitives were capable of delaying for less than the time requested, rather than possibly more.

This has been a long-standing feature of Linux, and most device driver developers deal with it by adding longer delays. Marc's point was that this can add up, especially in a flash driver of the kind he was developing. There was no resolution, but the discussion resulted in Linus documenting some of his expectations and assumptions around how Linux should be implementing time. ■



ULTIMATE
**RESCUE &
REPAIR KIT**

We can repair it, we have the technology. **Neil Bothwick** reveals the best tools for diagnosing, fixing, cleaning and maintaining systems



AT A GLANCE

- **Analyse problems, p20**

Look at the evidence to be sure you know what the problem is before trying various fixes.

- **Clone & back up, p22**

Make a copy of your data, recover it while you can and make sure any attempts at repair do not cause further loss.

- **Recover your system, p24**

What you can do to recover from the problem, with the two main objectives being

keeping your data safe and getting your system running again.

- **Hardware & cleaning, p26**

Diagnose faulty or failing hardware and clean up after yourself by getting rid of any packages and files that are no longer needed.

- **Maintenance tips, p28**

Keeping your system safe while also maximising your chances of recovery should your computer fail.

There can be a certain smugness around some Linux users regarding how their chosen OS is stable, secure and reliable; but stuff happens and it can happen to you. Power failures, disk drive errors, failing memory (the computer, not you) and even ageing PSUs can all cause failures, not to mention faulty or malicious software or plain old ID-TEN-T user errors.

There are things you can do to reduce the chances of failure, and other steps to mitigate the damage done when failure occurs, and we will look at some of them, but in the main we'll be exploring what to do when disaster strikes and how to recover from it as best you can.

Ask yourself, what is the most valuable part of your computer system? If your

answer refers to any item of hardware, you are almost certainly wrong. Hardware can be replaced; it may cost a few quid but it can be done and usually easily. That doesn't always apply to your data: purchased media can be downloaded or bought again, but personal photos, documents and emails can be lost forever.

Over the next few pages we will look at the steps involved in diagnosing and fixing problems, as well as ways of keeping your valuable data safe, both before and after the fact. While some issues can be fixed from a running system, even if it is running in a crippled way, others require the use of a live CD distro and we'll be using a couple of these along the way. The live distros are also on this month's cover disc, a very good

reason to keep it safe, but you can also boot them from a USB stick or even over the network if you prefer.

It is well known that one of the biggest causes of problems is solutions: ask a plumber how much he earns from clearing up after home fixes. When things go wrong, we need to take a structured and considered approach to the problem, being sure we know what is wrong before trying to apply, sometimes drastic, fixes. The good news is that there are plenty of Linux tools to help with this; there are even some that can be used to fix Windows problems, which we will look at briefly. There is also a lot you can do to avoid problems cropping up in the first place, from making regular backups to keeping an eye on your system's health. ▶



QUICK TIP

Read the man page

If you use `systemd`, read the `journalctl` man page. It may seem heavy-going at first, but it makes searching the system logs so much more effective.

Analyse problems

Before you go about trying to fix a problem, you will need to be sure precisely what that problem is

So what can we do to identify the problem without making it worse? The main thing to remember is 'don't guess'. Look at what happens when you experience the problem without trying to jump to the first possible cause that pops into your head. Does the computer boot? If not, at what stage does it stop and what messages do you see? Do you see a GRUB menu? Does selecting any option boot the computer? If the answer to any of these is 'no', reach for your copy of Rescatux which has various options to quickly fix GRUB issues.

Does Linux start to boot but never reach the desktop? Do you get as far as a login prompt in that case? These all help to pinpoint at what stage your system is failing. If you get as far as the distro starting to load services, you have a good chance of finding something in the logs. The same applies if the desktop loads but certain programs fail to run correctly.

Use the logs

Linux logs just about everything: kernel messages go to the kernel buffer while most other programs send information to the system log. If your distribution uses `systemd`, which most do now, these are combined in the journal. The `systemd` journal is basically a system log that has the added benefit of an index, which makes finding information much easier. With traditional loggers, the log is written to a plain text file which you can search easily enough with `grep` and other useful utilities. You can still use `grep` on the `systemd` journal, but there are better options.

This depends on the information being written to the log, and there are two situations where it is not. If

“The `systemd` journal is basically a system log but with an index, making finding info much easier **”**

the system halts early in the boot process, before the file system containing `/var` is mounted and the logger started, nothing can be written to the system log. Some initramfs implementations start the `systemd` journal very early, buffering the information in memory until it can be saved to disk, so if you can get to a login prompt, you can probably read the journal. If you don't get a prompt, first try entering single-user mode. At the GRUB boot menu (you may have to hold down a key when booting for it to appear), press `E` to enter edit mode and add `single` to the list of options on the kernel line. If you are using `systemd`, the option is '`systemd.unit=basic.target`'; if `basic.target` fails, try `rescue.target`. Also delete any options for `splash` or `quiet` – we want to see what the system has to say for itself – then press `Ctrl+X` to boot with those options. This should get you to a login prompt in single-user mode, the most basic of operating modes. Now you can check the system and kernel logs. The system log file is usually at `/var/log/messages`, so read it with:

█ `less /var/log/messages`

Some loggers use a different name, such as `/var/log/current`, but it should be one of the most recent files in `/var/log`. Read the kernel log with:

█ `dmesg | less`

Look though these for error messages. Don't worry too much about warnings: those can be generated by various conditions, like the system looking for something that isn't there, but warnings are not critical and shouldn't break things. With `systemd`, you use:

█ `sudo journalctl -b`

The `sudo` is necessary if you are not logged in as root as

```
Nov 22 18:40:46 phoucgh kernel: usb usb11-port4: cannot disable (err = -32)
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471561400
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944919, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944920, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944921, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944922, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944923, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944924, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944925, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944926, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944927, lost async p
Nov 22 18:40:46 phoucgh kernel: Buffer I/O error on dev sde1, logical block 58944928, lost async p
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471563448
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471565496
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471567544
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471569592
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471571640
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471573688
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471575736
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471577784
Nov 22 18:40:46 phoucgh kernel: print_req_error: I/O error, dev sde, sector 471579832
Nov 22 18:40:52 phoucgh kernel: usb usb11-port4: Cannot enable. Maybe the USB cable is bad?
Nov 22 18:40:52 phoucgh kernel: usb usb11-port4: cannot disable (err = -32)
Nov 22 18:40:56 phoucgh kernel: usb usb11-port4: Cannot enable. Maybe the USB cable is bad?
Nov 22 18:40:56 phoucgh kernel: usb usb11-port4: cannot disable (err = -32)
Nov 22 18:41:00 phoucgh kernel: usb usb11-port4: Cannot enable. Maybe the USB cable is bad?
Nov 22 18:41:00 phoucgh kernel: usb usb11-port4: cannot disable (err = -32)
Nov 22 18:41:04 phoucgh kernel: usb usb11-port4: Cannot enable. Maybe the USB cable is bad?
Nov 22 18:41:04 phoucgh kernel: usb usb11-port4: cannot disable (err = -32)
lines 89-117/179 65%
```

otherwise you may only see the log entries for your user. The **-b** option tells it to only show entries since the last reboot. You can also fine-tune it further by telling it to only show entries marked as error, or more serious, with:

```
sudo journalctl -b -p err
```

If you are more comfortable using a pager or grep to work with log files (journalctl is much more flexible but a broken system is not the best time to be learning about something you've not used before), you can send its output to a pipe and it will send plain text, like this:

```
journalctl -b | grep broken
```

Once you see an error, you should look for other messages relating to the same component. With the systemd journal, for example, if you see an error relating to **/dev/sda**, you can see all messages (system log and kernel buffer) about that drive with:

```
journalctl /dev/sda
```

QUICK GUIDE

GRUB options

There are a number of options you can add to the kernel line in GRUB to help when diagnosing a problem. First, remove the splash and quiet references so you can see what is going on. If you have already booted with a splash screen and the system just stops booting, press **Esc** to get rid of the splash screen and, hopefully, reveal something useful. Depending on the initramfs your distro uses, adding **rd.shell** and **rd.debug** may help. The former allows the system to drop to a shell if the initramfs fails to complete its handover, while **rd.debug** logs everything it does. This will be written to a file in **/run** or to the systemd journal if it is running, allowing you to inspect the log or save it to a USB stick for later perusal.

Sometimes the info you want to see flashes past too quickly for you to read it. You may be lucky and be able to get it back with **Shift+PgUp**, provided it hasn't gone too far. Normally, the kernel scrollback buffer holds only a few pages. Sometimes that won't work; the message you want has gone too far, the scroll isn't working or you simply haven't got as far as loading the kernel – as would happen with a GRUB error. One option is to record the screen with your phone's camera and play it back in slow motion until you see the information you need. It's a crude solution but it works.

When GRUB is working, it is unusual for anything to go wrong with it. Those errors that do occur are either missing file errors after an incomplete update or damage to the file system containing **/boot**.

```
journalctl -b -p err
[...]
0: 1 Mounted Kernel Configuration File System.
0: 1 Mounted Kernel Configuration File System.
0: 1 Mounted FUSE Control File System.
0: 1 Started Create Static Device Nodes in /dev.
0: 1 Started Create Static Device Nodes in /dev.
0: 1 Started Read Kernel Variables.
0: 1 Started Read Kernel Variables.
0: 1 Starting udev Kernel Device Manager...
0: 1 Started udev Kernel Device Manager...
0: 1 Started udev Kernel Device Manager...
0: 1 Reached Target Encrypted Volumes.
0: 1 Reached Target Local File Systems (pre).
0: 1 Mounted /tmp.
0: 1 Mounted /tmp.
0: 1 Started Set Console Font and Keymap...
0: 1 Starting Dynamic Variable Files and Directories...
0: 1 Started Set Dynamic Variable Files and Directories...
0: 1 Started Create Variable Files and Directories...
0: 1 Started Set Dynamic Variable Files and Directories...
0: 1 Started Set console font and keymap...
0: 1 Started Update UTPC about System Boot/Shutdown...
0: 1 Started Update UTPC about System Boot/Shutdown...
0: 1 Started Update UTPC about System Boot/Shutdown...
0: 1 Started Network Configuration...
0: 1 Started Network Configuration...
0: 1 Started Network Configuration...
0: 1 Listening on avahi-NEONS-SD Stack Activation socket...
0: 1 Listening on avahi-NEONS-SD Stack Activation socket...
0: 1 Listening on GND Daemon Activation socket...
0: 1 Listening on CPU Scheduler...
0: 1 Listening on GND Daemon Activation socket...
0: 1 Listening on G-Bus System Message Bus Socket...
0: 1 Started systemd-resolved-update-environment-path...
0: 1 Reached Target Paths...
0: 1 Started Message of the Day...
0: 1 Listening on D-Bus...
0: 1 Started Trigger execution every few minutes...
0: 1 Started D-Bus not upgrade and clean activities...
0: 1 Started Timer to automatically refresh installed snap...
0: 1 Listening on D-Bus activation for snapd daemon
```

QUICK TIP

See what went before

When you find an error in a log, check the entries immediately before the error: they may give a clue as to what led to it.

Left Turn off the pretty splash screen in GRUB and you can see exactly what is happening on your computer when it boots. All looks well here, but if there is a problem it should be easy to spot

However you get the log information, once you have it you can usually see what is going wrong, and once you know the cause of the problem you are on your way to the solution.

The other situation where information is not written to the log is when a program causes an instant crash, meaning nothing can be written to disk before the system goes down. Fortunately, such occurrences are rare and because the crash is instant, you usually know what you did to trigger it.

The system log or journal is not the only source of information. For example, if your system boots to

“When a program causes an instant crash, nothing can be written to disk before the system goes down”

a console login prompt when you expected to see a desktop, X is probably failing to load. In that case, look in **/var/log/Xorg.0.log**, in particular at any lines starting with '(EE)' – hint:

```
grep EE /var/log/Xorg.0.log
```

There are other logs files in **/var/log** – not everything uses the system log. Listing the directory in date order immediately after a problem occurs can reveal the most recently written log file, which may well hold what you need.

```
ls -lrt /var/log
```

Systemd users can use **journalctl -xe** to show more detailed information on the most recently logged events.

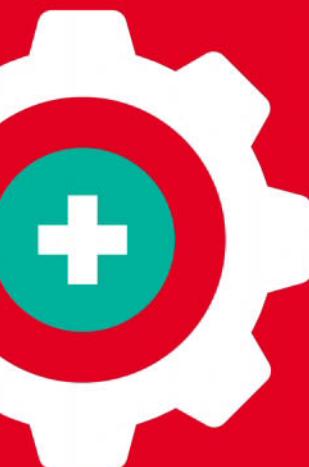
Applying these various techniques, you will either see an error message that you understand or one that baffles you. In the latter case, try pasting into your favourite search engine, but don't just jump on the first answer you see – make sure you read the results carefully to ensure you are trying the right fix for your situation. ▶



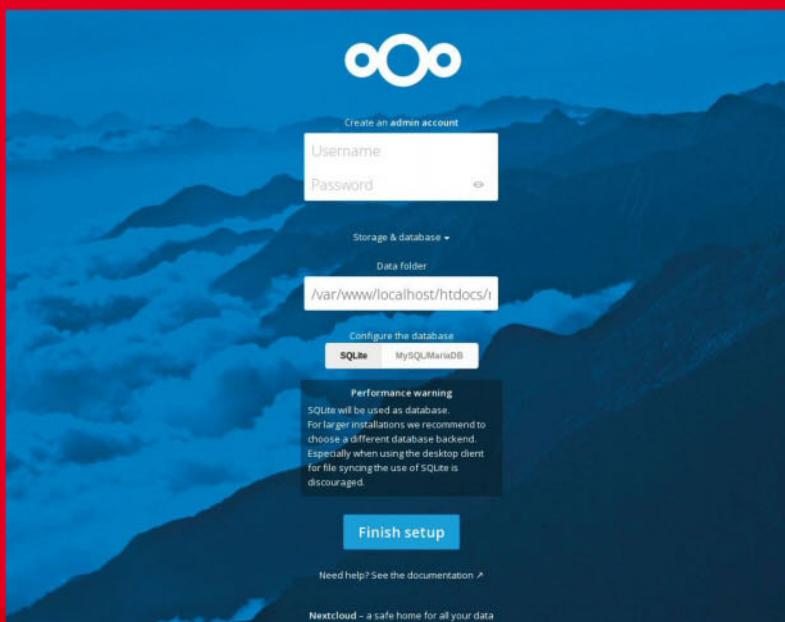
QUICK TIP

Store them
remotely

If possible, do not keep backups on the same computer, even if on a different drive. Back up to another computer on your network, or use a cloud storage service.



Below For extra security, send your backups to another location like an open source cloud server such as Nextcloud



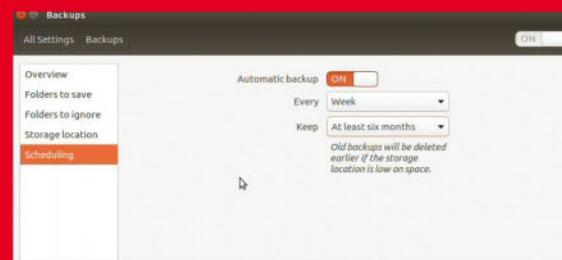
Clone & back up

Make a copy of your data, recover it while you can and make sure any attempts at repair don't cause further loss

If anything is wrong with a disk drive or file system, the easiest way to make it worse is to write to it. So the first step is to boot from a rescue CD so you can mount the drive read-only. Before mounting it, you should make a backup image of the drive. The usual tool for this is dd, but that program will exit with an error as soon as it encounters any issue reading from the drive, so we use ddrescue instead. This does the same job as dd but will try again if it hits an error and if it cannot read part of the drive, even after multiple tries, it skips on to the next part. Some of the tools we discuss can be used on an image file; in that case, make a copy of the image to work on – don't risk corrupting the first image in case you cannot make another one. You invoke ddrescue with three arguments:

```
ddrescue /dev/sda1 sda.img sda.map
```

The first is the device to read, the second is the image file and the third is a map file that ddrescue uses to keep track of any blocks it couldn't read. You can sometimes recover more data by running the command more than once: ddrescue uses the map file to know which blocks to try again, although this is more likely to be effective with scratched optical discs than hard drives. Naturally, you can't store either the image or the map file on the drive you are trying to image, so you need to use either



Above There are many backup programs available; this one is Déjà Dup. It is less important which one you choose, but essential that you actually use it, preferably on an automated schedule

a separate drive, maybe an external one, or a network mount. If your copy stops because it can't read past a damaged sector of the source drive, run it again with the -R option, which scans the disk in reverse. If there is only one sector that is damaged, this should reconstruct all of the disk apart from this part. If there are multiple occurrences, investigate the -i option that allows you to scan from a specific position.

The ddrescue manual at <http://bit.ly/ddrescuemani> contains a number of useful examples, including one with details on handling this situation.

Backups

Much grief can be avoided by making regular backups. As well as a means to recover data in the event of a loss, backups are important in the recovery process.

“There is always the risk of any attempt to repair file system damage actually making things worse”

Whenever there is file system damage, there is always the risk of any attempt to repair it actually making things worse. We've already covered ddrescue, but there is also Clonezilla. This can make a complete image backup of a hard drive, including boot sector, partition table, Linux and Windows file systems. Clonezilla is a live CD, available from <http://clonezilla.org>, that can clone a drive to an image file, but it can also clone it to another drive. This is a good option if you have a drive

that's failing. Clone the old drive to the new one, then you can check the integrity of the data on the new one without disturbing or writing to the suspect drive. When copying to an image file, Clonezilla can store it on another computer using a variety of network sharing and cloud protocols, but it can also use plain old SSH.

Backups are also important from a prevention point of view. There are two important points to remember about backups. The first is that humans should never be trusted with doing them. Most backup programs have an option to schedule regular backups; this is how to do it. There are many to choose from. Your author uses Duplicity, a command-line program that can be from cron, but there's also a GUI front end called Déjà Dup which makes operation much simpler on desktop systems. These can back up to an external hard drive, another computer via SSH or to many cloud services.

The other important point about backups is that you must test them. Most backup programs will let you do a test restore to a different location, just to be sure that the files you thought you backed up are actually there and able to be restored. When you have just lost the originals is not the time to find this out.

“Most backup programs have an option to schedule regular backups; this is how to do it **”**

QUICK TIP Keep them separate

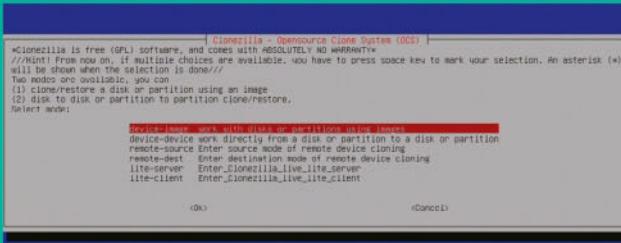
Keeping your OS and data on separate partitions can limit the damage cause by a failure. Losing either your OS or home is bad, but not as bad as losing both.

use it, do a real restore to a different location, just to be sure that the files you thought you backed up are actually there and able to be restored. When you have just lost the originals is not the time to find this out.

If your data is really important, you should back it up to a different location: you don't want a power surge destroying both the original and backups. With cheap cloud storage and fast internet connection, this is more practical than ever. Which data is important? If you ever have a drive failure without backups, you will realise it all is. ▶

HOW TO

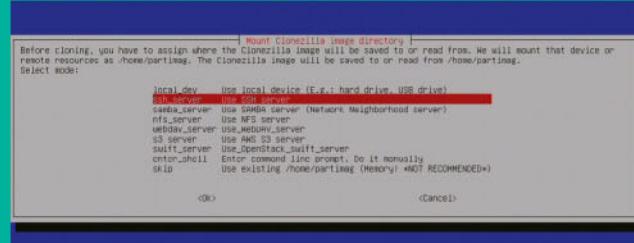
Clone a disk using Clonezilla



1

Choose a method

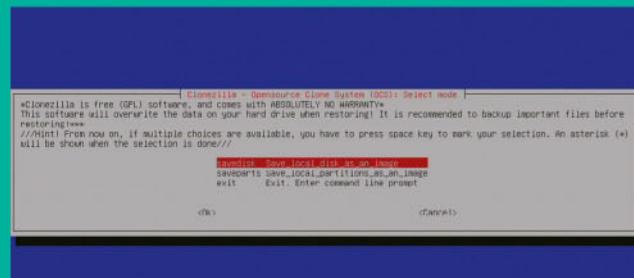
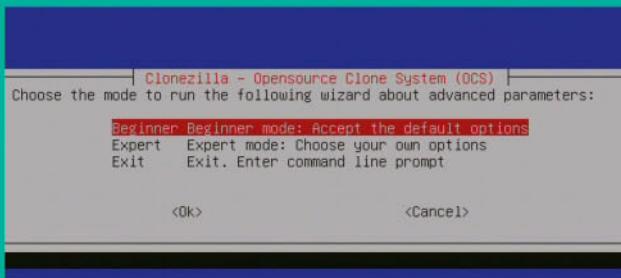
Boot from the Clonezilla live CD and choose a cloning method. Here we are sending to an image file, but you can also clone to another disk, as long as it is at least as large as the source drive. Other options allow for a network backup to or from a Clonezilla server running on another computer.



2

Storage locations

When backing up to an image file, it does not have to be on the same computer. Backing up to a second local drive is fastest, but you can also use a number of network protocols. SSH is the simplest when it can be used, but you can even backup to a web server using WebDAV.



3

Beginner or Expert?

Choose your level of expertise. Beginner mode makes the difficult choices for you and is the right option most of the time. Expert mode enables you to tweak several of the backup settings, which may be good for Clonezilla gurus but just gives everyone else more chances to get it wrong.

4

Whole or partition?

Clonezilla can save a whole disk or just a single partition. When saving a whole disk you get the partition tables, bootloader and an image of each partition, plus the info Clonezilla needs to put it all back together. A partition backup is just a compressed image of that partition.

QUICK TIP

Don't get frazzled

Recover your system

What you can do to recover from the problem, keep your data safe and get your system running again

Disk drive failures are a common cause of grief and can fall into a number of categories.

Terminal drive failure is the worst and you can usually say goodbye to your data, unless it is valuable enough to justify the services of a specialist recovery company, but anything that valuable would be backed up, right? Next is a corrupted partition table, where the drive is visible but none of its partitions can be seen. Before you start trying to fix the drive on these cases, may sure it is at fault. Try a different cable, plug it into a different port, or even computer, to be sure that it is not something as simple as a loose or damaged cable, or a problem with your computer's disk controller hardware. Naturally, this is more difficult with a laptop system but still possible in most cases. Another category is file system errors: the partitions are visible in `/dev` but cannot be mounted, or will mount but the data appears garbled. Finally, we have missing files when the file system is intact but important files have been deleted (by someone else of course).

Where we go next depends on the type of fault with the drive. If the drive shows up but the partitions don't, or the partition table is otherwise corrupted, we can rebuild the table. See the instructions at the bottom of the page.

Corrupted file systems

If the file system mounts but the data or metadata looks wrong, you probably have file system corruption, possibly

caused by an unclean shutdown. The standard tool for fixing this is `fsck`. At its most basic, you invoke:

```
| fsck /dev/sda1
```

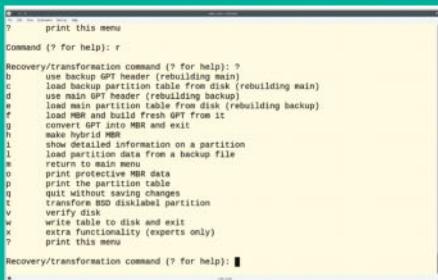
Note that you run **fsck** on a partition, not the whole disk. The **fsck** program is basically a wrapper that identifies the file system type and then calls the **fsck** program for that type, for example **fsck.ext4**. For some file systems, the **fsck** program actually does nothing but returns success. For those, such as **XFS**, you will need to run the appropriate repair tool. For standard file systems like **etc3/4**, **fsck** should do the job. It is an interactive program and will prompt for each error or inconsistency it finds. These could run initially thousands, so you may want to run **fsck** with the **-y** option, which assumes a yes response to most questions, but not the most critical ones. Alternatively, use **-p**, which fixes any problems that can be safely repaired without intervention.

If your system won't boot, or you have some sort of hard drive error, you should boot from a live CD to carry out any further work. There are many live Linux CD/DVDs, most of which are generally useful. If you have something like the latest Ubuntu image hanging around, that may well have all the tools you need.

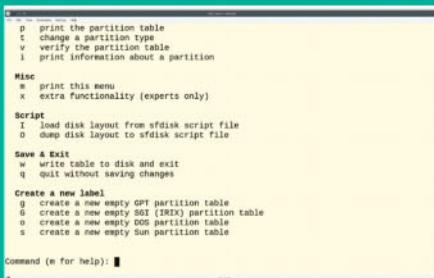
However, there are live distros specifically designed for use when problems arise, such as SystemRescueCd,

HOW TO

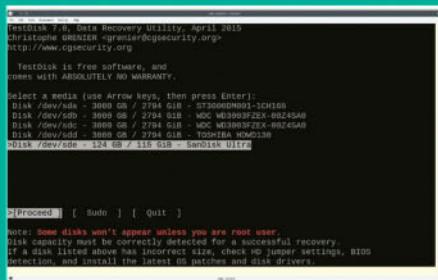
Recover lost partitions with Rescatux



1 Rebuild GPT partition table
The GPT standard stores backup copies of the partition table on the disk. Using gdisk, press **R** to enter the recovery menu and **C** to load the backup table. To rebuild the main partition table from the backup, press **B**, then **W** to save.



2 **DOS partition table**
The old-style DOS partition table is less friendly, with only one copy. So before things go wrong, use fdisk to dump the partition data to a file, just in case things go wrong. If you don't have a backup, you'll need to try testdisk.



3 Scan for partitions
You can use testdisk to scan the entire disk for the tell-tale signs of partitions. Run testdisk from a root terminal to see a list of available drives. You can also pass it the name of an image file you created with dd or ddrescue.

which boots to a command prompt by default but can also be booted to a rather spartan-looking desktop if you're more comfortable with that. As is implied by the name, this distro contains many of the tools needed to repair Linux (and Windows) systems. It's on this month's cover disc and it would be a good idea to keep this in a safe place. The distro also includes a script to install itself on a USB drive for a more portable solution.

Another popular rescue disk is Rescatux. While the range of problems that can be solved with this one is more limited, the one it is designed to fix can usually be resolved by the click of a button. Its RescApp program contains a number of buttons to fix common problems, making it good for a quick fix.

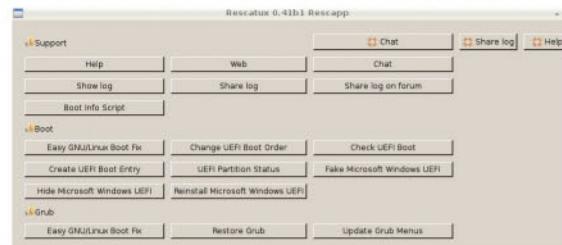
Not all faults are caused by hardware or software; quite a few are caused by users. Hands up anyone who's never forgotten a password – they're the ones with weak passwords. Forgetting a user password is not usually an issue if you can log in as root to change it, but what if you forget the root password, or need to use sudo as on Ubuntu systems? It's actually remarkably easy: after booting into a rescue CD, mount your root partition at, say, `/mnt/root`, then change the password with `passwd`:

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

Rescue CDs log you in as root when they boot, so you can change the password of any user, including root, using this method.

Linux to the rescue

While using a Linux live CD to fix a broken Linux system seems entirely logical, using one to fix Windows problems is not so immediately apparent, but it is possible. The obvious application of this is mounting an unbootable Windows partition in order to recover your documents and other files. NTFS support in the Linux has always



Left It may not be pretty, but Rescatux does make fixing certain problems, for Linux and Windows, very easy. Click the button, read the help, click Run and all is well!

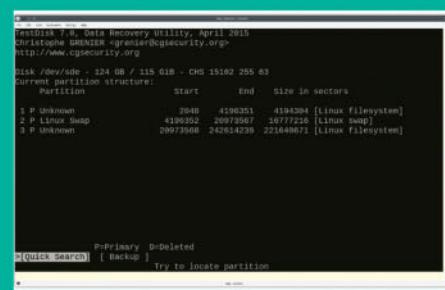
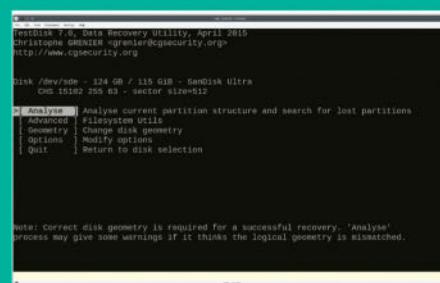
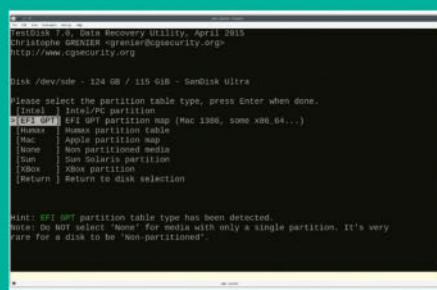
been rather limited, but it does allow read-only mounting of Windows partitions with no issues. Alternatively, you can use ntfs-3g, a FUSE file system that allows full read-write access. Given the previous comments about not trying to write to a broken system, this may not be such an advantage in this case, but most distros and live CDs include ntfs-3g and you can mount a Windows partition with:

```
ntfs-3g -o ro /dev/sda1 /mnt/windows
```

This uses the option to mount read-only, which is recommended if Windows did not shut down cleanly. The mount point, `/mnt/windows` in this example, must exist.

Also included with the ntfs-3g package is ntfsfix, which is run with the Windows device as its only argument. It fixes some NTFS problems, resets the journal and forces a check of the file system next time you boot into Windows, but it is not a Linux version of `chkdsk`.

You can also fix lost Windows passwords. Both SystemRescueCd and Rescatux can do this for you. With SystemRescueCd, go to the system tools submenu at the boot screen. Rescatux makes it easier: there is a set of buttons for Windows operation in RescApp, including password resets and promoting a user to administrator – both of which can help when you have locked yourself out of Windows. ▶



4 Choose format

You need to know the type of partition table to use. Most distro (and Windows) installers still use the Intel/DOS format, so select that unless you know otherwise. This is the format testdisk will use to save back any partitions it finds.

5 Look for boundaries

Analyse is where the work really starts. Testdisk will scan your entire disk looking for signs of partition boundaries. Depending on the partition table corruption, whether there's a backup and the disk size, this may take a while.

6 Back up info

Once partitions are detected, testdisk will offer you the option to back up the current information before going ahead and attempting further recovery attempts. We strongly recommend that you do this!

QUICK TIP
Get your duster out

Keep your computer clean. If it gets dusty on the outside, think what the inside may be like. A couple of minutes with an air duster every so often is better than waiting for problems to strike.

Hardware & cleaning

Diagnose failing hardware and clean up after yourself by eliminating packages and files that are no longer needed

While we are looking at the use of software to fix things, sometimes the problem exists in hardware, so we need to be able to test it.

Random crashes or failures are particularly difficult to diagnose, but there are a number of things you can look for. Was the computer heavily loaded at the time? Were the fans making more noise than usual? Was there a lot of disk access? Most hardware is really reliable these days, but there are a few weak links. The first is the power supply. Cheap, unbranded power supplies can age badly and cause random failures as voltages fluctuate. They are also difficult to test thoroughly without proper test equipment, but there are a couple of simple tests you can try. If you have a desktop computer with a standard ATX power supply, and a spare available, just swap it out. That is the easiest way to test. Another option is to unplug non-essential hardware, to reduce the load on the PSU. If it has just reached the point where it is causing failures, a slight reduction in load can bring it back into its comfort zone.

The next cause of instability is heat. Modern hardware contains multiple temperature sensors, so install a system monitor that lets you keep an eye on them. Some distros use Conky, which can be very attractive. Or you can use GKrellM, which is less attractive but much easier to set up to tell you what you want to know. Computers rely on a good flow of air to keep them cool; in the case of laptops, often through small spaces. It is easy for this path to get partially blocked, reducing cooling and increasing temperature. A can of compressed air (often called an air duster) is a good way of clearing things out – whatever you do, don't be tempted to use a vacuum cleaner as they can generate enough static electricity to damage your electronics.

Below While smartd monitors your drive's health in the background, you can use GSmartcontrol to view its state graphically

QUICK GUIDE**Power protection**

One of the easiest ways to shorten the life of hardware is with dirty power. A decent UPS (uninterruptible power supply) not only allows your computer to shut down cleanly in the event of a power failure, it protects against voltage fluctuations (brownouts) that can seriously weaken hardware, especially disk drives, and voltage surges that can harm motherboard components and more. One UPS can protect more than one computer.

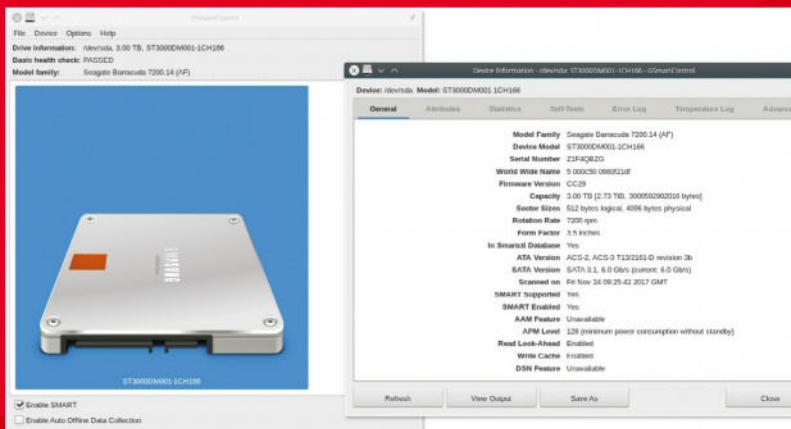
Power supply and cooling problems are relatively easy to fix, but if left unattended can cause expensive damage. Fluctuating voltages and high temperatures can damage expensive components. A new power supply can even save you money as modern ones are more efficient, saving you on your power bills and reducing the overall temperature load.

A failing power supply can make the computer more likely to fail when under heavy load, and so can faulty memory. The good news is that this one is easy to test. Many boot discs, including our DVDs, have an option to run memtest86 (or its fork, memtest86+). Memtest86 runs a battery of tests on your memory and reports any errors. As memory failures can be intermittent, you should let it run for at least two or three passes, overnight is ideal. If a failure shows up, you will need to remove individual memory sticks and run the test again to see which one is faulty – unless you have only one to start with.

Getting SMART

The other weak link in the hardware chain is the hard drive. While some people worry about the lifespan of SSDs, the truth is that they should last for years with no problems. Hard drives, on the other hand, are complex electromechanical beasts with lots of parts to go wrong. Luckily for us, they are also easy to test thanks to a technology called SMART (Self-Monitoring, Analysis and Reporting Technology) built into most drives. You may need to enable SMART in the computer's BIOS setup menu, then you can use a program called smartctl to check the drive's health. Unlike testing memory, most SMART tests run in the background while the drive keeps working as normal. To get a health report on a drive, run:

```
sudo smartctl -H /dev/sda
```



We use sudo because smartctl needs root permissions for low level access to the drive. To run a thorough test on a drive, you would use:

```
sudo smartctl -t long /dev/sda
```

This test can take a while and smartctl should give you an estimate of the time needed. When it is complete, you can see the results with:

```
sudo smartctl --log=short /dev/sda
```

And see details of any errors found with:

```
sudo smartctl --log=error /dev/sda
```

This does rely on you running the program regularly and looking at the log output, something we humans are really bad at. Fortunately for us, smartctl has a sister program, smartd, that runs in the background. Just edit `/etc/smartd.conf`, find the line that contains only `DEVICESCAN` and change it to:

DEVICESCAN -m me@my.email.com

Clean

A clean, tidy system is less likely to experience problems. We are using two meanings of 'clean' here: free of infection and free of clutter. Computers are good at many things; one thing they do particularly well is accumulate unwanted files all over your hard drive. Apart from wasting space, causing fragmentation and increasing backup times, this can have an adverse effect on your system. You can clean out your **home** directory from time to time, getting rid of those files you downloaded but no longer need. There are also temporary files that various programs save to your disk: things like browser cookies, cached files and various other cruft spread around. Help is at hand in the form of BleachBit (www.bleachbit.org). This program scans your **home** directory and identifies files that you should no longer need. It can take a while to complete its magic, then it shows you a list of what it has found and you can choose which files it should delete.

And cleaner

Having unnecessary programs installed is also a potential risk to your system. Those programs you installed to try out then never used again are still there, and some can present a security risk. Uninstalling unwanted programs is a start, but often when you

install a program, it pulls in some dependencies. With Debian/Ubuntu-based systems, you can get rid of dependencies that are no longer needed with:

```
| sudo apt-get autoremove
```

Keeping clean from infection is even more important. While Linux is considered by many to be immune to virus contamination, that is not true. While there has been little Linux malware, there has been some. If you share files with Windows systems, you also need to be aware of the possibility of transferring Windows viruses, which are a far more common beast. While Linux has very little in the way of viruses, it does have a good virus checker, ClamAV. ClamAV should be in your distro's repositories and consists of a number of programs. The

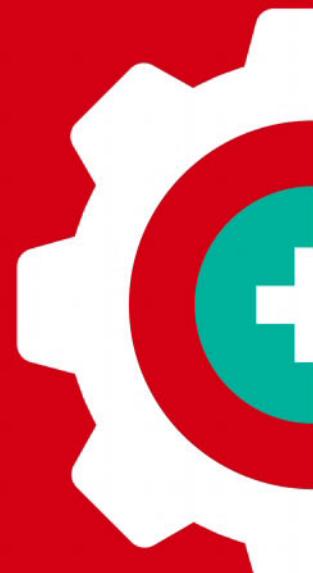
“ While Linux has very little in the way of viruses, it does have a good virus checker, ClamAV ”

first you really need is `freshclam`, which updates the virus databases – run this before scanning, or from a cron job. You should then run `clamscan` to check for viruses, giving it the directory to search, for example

```
clamscan --recursive --cross-fs=no /
```

The recursive option descends into subdirectories, so this would scan the whole file system. Disabling cross-fs prevents it searching other file systems, like **/proc** or any network file systems. It also means that if, for example, **/home** is on a separate partition, it will another invocation of clamscan. If your computer is always on, it is easier to drop a two-line script into **/etc/cron.daily** that calls freshclam and clamscan to check everything every night. ▶

Above BleachBit can find an amazing amount of cruft, but make sure you really don't need any of these files before letting it delete them



Left Testing memory with memtest86+. With a name like that, what else would you use it for? It is important to let it run for several passes – overnight is usually best

QUICK TIP
Don't get a complex

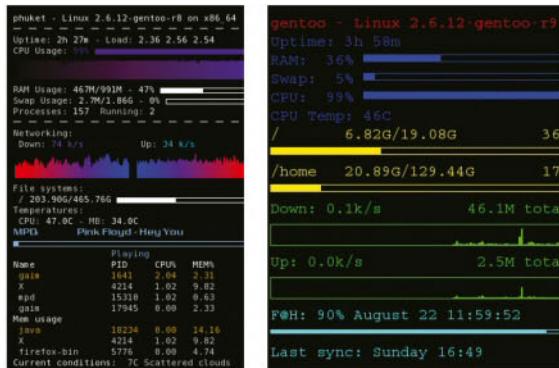
Keep your computer safe, but don't get hung up on it. Remember to spend most of your time using it for what you bought it for.

Right Conky is a highly configurable system monitor that enables you to keep an eye on many aspects of your system's health and status

Maintenance tips

Keep your system safe and prepared for problems to maximise your chances of recovery should it fail

Once everything is working well after you have repaired it, or preferably before you have needed to, there are things you can do to keep your system in good health, or at least be better prepared for problems. The first, and most important, is to set up a backup system, and make it an automated one. Don't rely on your remembering to do it, but do test the backups from time to time. While you are automating the backups, set up **smartd** to run in the background and have cron run Rootkit Hunter. Once those are set up, you can forget about them: they will let you know if they want your attention.



To keep an eye on your temperatures: a desktop system monitor like Conky or GKrellM can help here, showing you immediately if things start to get warm. You will naturally see temperature increases when your computer is working hard, transcoding a video or compiling software, but if it starts to rise in normal use you should investigate immediately. High temperatures can kill hardware, but don't wait for them to get too high: more modest rises can shorten the life of components. Laptops in particular can suffer from borderline cooling – are the fans louder than before? In most cases it is simply a matter of cleaning dust from around cooling vents.

Tidy up after yourself

If you often try different software packages, remove the ones you opt not to use. The more software installed, the greater the chance of exposing your system to a security vulnerability. Similarly, keep your system up to date: most updates to software are to fix bugs or security. Try to keep to packages from your distro's main repositories as much as possible, as they are tested for compatibility. Adding loads of PPAs may seem a good idea, but you may (not will, only may) end up with a less stable system.

Never use the power switch to shut down your system.

QUICK GUIDE

The magic of SysReq

Linux is usually stable enough to survive misbehaving software, but it is possible for a program to lock up the whole computer. Before you reach for the Power or Reset button and risk corrupting your file systems, there is a better way to get out of it. You can send commands directly to the kernel by holding down the Alt and **SysReq** (aka **PrtScr**) keys and pressing certain letter keys. As the kernel listens for these directly, they work even if X is completely locked and accepting no input.

The keys normally used to get out of a lockup are **R**, to reset the keyboard – this occasionally fixes the problem on its own. Next, press **E** to send a TERM signal to all processes, asking them to shut down cleanly, writing any data to disk and closing any open files. Next in line is **I**, which sends a KILL signal to all remaining processes, forcing them to shut down. Pressing **S** tells the kernel to sync, flushing all buffers to disk so that remaining open files can be closed cleanly. Press **U** to unmount all file systems and remount them read-only, to avoid further data corruption. Finally, **B** reboots the system.

So, that is: hold down Alt and **SysReq** and press **R-E-I-S-U-B** in turn, preferably leaving a couple of seconds between each. There are several, mainly silly, mnemonics to help remember that sequence, the most appropriate being Reboot Even If System Utterly Broken, but the easiest way to remember the sequence is that it is **BUSIER** backwards. This is not something you should need very often, but it is well worth remembering when you do.

If your desktop locks up and appears unresponsive, it may still be possible to reboot your computer cleanly – cutting the power is asking for a corrupted file system. If it is only the desktop that is locked up, and you're running an SSH server, try logging in from another computer and issuing the **shutdown** command from there (or **systemctl reboot** on a systemd distro). If all else fails, try the magic SysReq key combinations: these operate at the kernel level and can help you shut down cleanly when nothing else responds (see The Magic of SysReq box for details).

Most importantly, though, don't panic. Most of the time your system will just work. If it should suddenly hit problems, there is almost always something you can do about it, and it should only involve some common sense and detective work, not a credit card. ■

IPHONE X OR PIXEL 2? FULL TESTS



T3
SMARTER LIVING

THE BEST TECH

Christmas GIFT GUIDE

Over 190 brilliant buys,
from £20 to £8,000

4K TVs UNDER £700

Top new tellies that
don't break the bank



TESTED!

Wireless
smart home
security
cameras



6 ways to get
perfect Wi-Fi
everywhere

AMAZON FIRE TV 4K
VS. APPLE TV 4K!



WIN!
Over £9k
of tech!

EXCLUSIVE!
MEET THE TECH
THAT POWERS
Disneyland

PLAYED!
XBOX
ONE X

4K, HDR, Dolby
Atmos – it's the
future of gaming



Future
ISSUE 278 DEC 2017
PRINTED IN THE UK £4.99
Barcode
9 771364 264131
12>

ON SALE NOW!

AVAILABLE AT WHSMITH, MYFAVOURITEMAGAZINES.CO.UK
OR SIMPLY SEARCH FOR T3 IN YOUR DEVICE'S APP STORE

SUBSCRIBETODAYANDSAVE! SEE WWW.MYFAVOURITEMAGAZINES.CO.UK/T3



Subscribe

Never miss an issue

FREE DVD TRY 2 UBUNTU SPINS

FREE DVD! 4.7GB OF DISTROS & TOO

LinuxUser

THE ESSENTIAL MAGAZINE
FOR THE GNU GENERATION



> 100 WAYS TO
MASTER THE
TERMINAL

> Become a command line guru in 24h

INTERVIEW
ZeroPhone

The hero building a \$50
phone with the Pi Zero



EXPERT GUIDE

Top tips to
stop your
passwords
from being
cracked



£6.49
£5.19
per issue

LinuxUser

www.linuxuser.co.uk

THE ESSENTIAL MAGAZINE
FOR THE GNU GENERATION



ULTIMATE
RESCUE &
REPAIR KIT

Digital forensics • Data recovery • File system
repair • Partitioning & cloning • Security analysis

2 TOP DISTROS TO TRY NOW!

ANTERGOS
Modern, elegant and powered by Arch



XUBUNTU
Based on Ubuntu and built for speed



+ RESCUE & REPAIR TOOLS
SYSTEMRESCUECD
AND RESCATUX

Salvage your system with two great distros

+ Go deep into network data with WIRESHARK



42 PAGES OF
GUIDES

> MQTT: Master the IoT protocol

> Security: Intercept HTTPS

> Essential Linux: The joy of Sed



Ubuntu 17.10 SPECIAL

THE ESSENTIAL MAGAZINE
FOR THE GNU GENERATION



MASTER
UBUNTU
17.10

All the new features!
PLUS the best extensions,
customisation tools and
alternative spins

NEW RELEASE

Install it ✓ Customise it
✓ Master it



Pinebook 64-bit

ALSO INSIDE

> Pi status board: monitor everything
> Jupyter: write and share live code
> Build a social media holiday sweater

9 77204 527019

Subscribe and save 20%*

Every issue, delivered straight to your door



Never miss an issue

13 issues a year, and you'll be
sure to get every single one



Delivered to your home

Free delivery of every issue,
direct to your doorstep



Get the biggest savings

Get your favourite magazine for
less by ordering direct

What our readers are saying about us...

"I've only just found out about this
magazine today. It's absolutely brilliant
and exactly what I was looking for.
I'm amazed!"

Donald Sleighholme via Facebook

"@LinuxUserMag just arrived by post.
Wow what a fantastic issue! I was just
about to start playing with mini-pcs and
a soldering iron. TY"

@businessBoris via Twitter

"Thanks for a great magazine. I've been a
regular subscriber now for a number
of years."

Matt Caswell via email

Pick the subscription that's right for you



Subscribe and save 20%

- ✓ Automatic renewal – never miss an issue
- ✓ Pay by Direct Debit

Recurring payment of £33.75 every six months, saving 20% on the retail price



One year subscription

- ✓ Great offers, available worldwide
- ✓ One payment, by card or cheque

A simple one-off payment ensures you never miss an issue for one full year. That's 13 issues, direct to your doorstep



Instruction to your Bank or Building Society to pay by Direct Debit

Originator's reference

7 6 8 1 9 5

Name of bank

Address of bank

Account Name

Postcode

Sort Code

Account no

Please pay Future Publishing Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit guarantee. I understand that this instruction may remain with Future Publishing Ltd and, if so, details will be passed on electronically to my Bank/Building Society. Banks & Building Societies may not accept Direct Debit instructions for some types of account

Signature

Date

UK £67.50 (saving 20% on the retail price)

Europe €88.54

USA \$112.23

Rest of the world \$112.23

Pay by card or cheque

Pay by Credit or Debit card



Visa



Mastercard



Amex

Card number

Expiry date

Pay by Cheque

I enclose a cheque for

£

Made payable to Future Publishing Ltd

Signature

Date

Your information

Name

Address

Telephone number

Mobile number

Email address

Postcode

Please tick if you want to receive any communications from Future and its group companies containing news, special offers and product information.

Please post this form to

Linux User & Developer Subscriptions, Future Publishing Ltd, 3 Queensbridge, The Lakes, Northampton, NN4 7BF, United Kingdom

Order securely online www.myfavouritemagazines.co.uk/sublud

Speak to one of our friendly customer service team

Call **0344 848 2852**

These offers will expire on

31 January 2018

Please quote code **LUDPS17** when calling

*Prices and savings are compared to buying full priced print issues. You will receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: bit.ly/magtnode Offer ends 31 January 2018.



John Gowers

is a university tutor in Programming and Computer Science. He likes to install Linux on every device he can get his hands on, and uses terminal commands and shell scripts on a daily basis.

Resources

■ A terminal running the Bash shell (standard on any Linux distro)

■ A 'dictionary' text file, containing a line-separated list of words in the English language (check the /usr/share/dict and /var/lib/dict directories or search online)

■ The wget program (see your package manager or download from www.gnu.org/software/wget/)

PART FIVE

Master shell scripting: The Joy of Sed

Learn the many uses of the stream editor sed, an extremely versatile command for text replacement

In the first part of this series, we learned about the UNIX philosophy that underpins all the commands we have been learning about. Part of that philosophy was the insistence that all programs should communicate with one another through a text stream, since that is a universal interface. This is a powerful idea: as long as two programs function by taking in textual input and printing out textual output, then they can interact together through a pipeline, even if they haven't been explicitly designed to work together.

The main pitfall behind this idea is that the textual output from one program might not be in the format that the second program expects as input. In the last issue, we met one of the tools that can be used to get round this program: the program **grep** filters out lines of output depending on whether they fit a particular pattern.

In this issue, we will meet a more powerful program: the 'stream editor', or **sed**, that can act as the glue between different programs. Rather than filtering lines based on a pattern, **sed** can actively change a line

of output through applying substitutions. **Sed** is the precursor of more powerful stream editors such as AWK, Perl and Python; it is very well suited to simple text-processing tasks. For example, we will later see that we can use **sed** to translate the output of a package manager program into input accepted by the **rm** program.

s is for substitute

The most important **sed** command, and the one we will use most frequently, is **s**, which is short for 'substitute'. The basic syntax is very simple:

```
$ <<< "couscous" sed 's/cous/pom/g'  
pompom
```

In this example, the **s** at the beginning is the command, substitute, followed by a slash /. This is followed by the string to remove, **cous**, followed by another slash / and lastly by the string to replace it with, **pom**, followed by a

```
#!/bin/sed -f  
  
/\n/ !G  
s/(. )\(. *\n\)/&\2\1/  
//D  
s/. /
```

	olleH
/\n/ !G	olleH\n\$
s/(.)\(. *\n\)/&\2\1/	olleH\nlleH\no\$
// D	lleH\no\$
s/(.)\(. *\n\)/&\2\1/	lleH\nleH\nlo\$
// D	leH\nlo\$
s/(.)\(. *\n\)/&\2\1/	leH\nneH\nllo\$
// D	eH\nllo\$
s/(.)\(. *\n\)/&\2\1/	eH\nH\nello\$
// D	H\nello\$
s/(.)\(. *\n\)/&\2\1/	H\n\nHello\$
// D	\nHello\$
s/(.)\(. *\n\)/&\2\1/	\nHello\$
// D	\nHello\$
s/. /	Hello\$

third slash /. Last, the letter g is a supplementary option to the s command: it tells s to apply the substitution throughout the line. If we leave off the g, then sed only applies the substitution once per line:

```
$ <<< "couscous" sed 's/cous/pom/'  
pomcous
```

The power of the s command comes from the fact that the first argument can be an arbitrary regular expression, rather than just a string. For example, we can write a Sed script that will delete all non-alphanumeric characters from input:

```
$ sed 's/[^A-Za-z0-9 ]//g'  
It is time to eat, grandpa.  
It is time to eat grandpa
```

If we want to use the Extended Regular Language, then we can use the -E switch, just as with grep:

```
$ sed -E 's/[aeiouwy]+/oodle/g'  
Do it! Now!  
Doodle oodlet! Noodle!
```

Here, the extended regular expression [aeiouwy]+ matches any sequence of vowels (or ws or ys).

Grouping with sed

In the last issue, we learned that we can group parts of a regular expression together using brackets preceded by backslash (\(...\)) or, in the Extended Regular Language, plain brackets ((...)). grep then allowed us to refer back to these groups using numerical codes \1, \2 and so on. For example, the following regular expression matches a word, followed by a second word, followed by the first word:

```
$ grep -E '([^\s]+) \1'  
hand in hand  
hand in hand  
hand in glove
```

Since we have enclosed the pattern [^\s]+ corresponding to the first word in brackets, we can refer to it later on as \1.

We can use exactly the same notion of grouping for our regular expressions with sed. Backreferencing of groups is even more powerful in sed, however, since we can also refer to the group on the right of the s command. This allows us to refer to replace a particular string with a

“Text-processing programs such as sed are the glue between different command-line programs”

FLAGS TO THE S COMMAND

g	Apply the substitution wherever it appears in the line
3	(or another number). Apply the substitution the 3rd time it appears in the line
p	Print out the substituted string an extra time (often used with the -n flag to suppress automatic printing)
w filename	Print the substituted string to the specified file
e	Treat the substituted string as a Linux command and replace the string with the output of that command

second string that is derived from the first. For example, we can write a sed script to reverse the first two words in each line:

```
$ sed -E 's/([^\s]+) ([^\s]+)/\2 \1/'  
is sed worth learning  
sed is worth learning
```

As with grep, the numbers 1, 2 and so on refer to the order in which the bracketed expressions appear in the matching regular expression. In this case, we want to print the second group first and then print the first group.

A related expression in sed is the ampersand &. If we use this on the right-hand side of an s command, then it refers to the expression that was matched on the left:

```
$ sed -E 's/([aeiou ]?)([^\s]*)/& in pig latin  
is \2\1ay./g'  
hello  
hello in pig latin is ello-hay.  
apple  
apple in pig latin is apple-ay.
```

Chaining sed commands

Sometimes, we might want to run more than one sed command over our program in sequence. Sed allows us to chain commands together using the semicolon character ‘;’:

```
$ sed 's/dog/dawg/g;s/cat/katt/g'  
It's raining cats and dogs.  
It's raining katts and dawgs.
```

Note that each command is run over the string in turn. So the command s/cat/dog/g;s/dog/cat/g will not swap the words cat and dog: it will first replace all instances of cat with dog and then replace all instances of dog with cat. The result is that the original instances of cat are changed back to cat again.

Sed scripts

Just as we wrote Bash scripts before, we can write sed scripts, which can be run as standalone programs.

Above The s command provides many more flags than the ones we have met. To use these, put them at the end of the command. For example, s/dog/cat/3

■ Escape characters and single quotes

As we learned in part two of this series, texts in single quotes are treated exactly as they are written, whereas strings in double quotes are subject to various manipulation rules. It might seem surprising that, although sed supports escape characters such as \n for a new line, sed scripts are almost always written to the command line in single quotes. The reason for this is that it is sed itself that interprets the escape characters: if we put the string in double quotes, they'd be interpreted by Bash and would then not work correctly with sed.

Further directions

The commands we have learned are quite simple, but sed supports further commands that allow us to perform looping, saving of state and operations on multiple lines at a time. If you want to learn about these, a good place to start is by typing `man sed` or (better!) `info sed` at the command line.

The sed script in the main image is an emulation of the `rev` command, which reverses its input. If you want to understand some of the more advanced features of Sed, a good place to start is by trying to convince yourself why this script works as it does.

Right Sed scripts are used in many major pieces of software. This one is part of Bash itself, used for printing quotation marks

To write a sed script, we start off with the following shebang line:

```
#!/bin/sed -f
```

...which tells the shell that the script should be run by the `sed` command. If we want to use the Extended Regular Language, we should use `#!/bin/sed -Ef` instead. We can then list the sed commands one by one. For example:

```
#!/bin/sed -f

# Remove spaces
s/ //g
# Replace 'gobi' with 'tuna'
s/gobi/tuna/g
```

Save this into a file called `luad-replace.sed`, and then run `chmod +x luad-replace.sed`. You can then run the script directly from the command line:

```
$ ./luad-replace.sed
forgo bite
fortunate
```

Note that we used comments in the script above. In a sed script, any file starting with a hash sign `#` is treated as a comment and ignored by `sed`.

Editing files with sed

It is easy to use the normal pipeline to run a sed command over a file. For example:

```
$ <email.txt sed 's/damn/darn/' > clean_email.txt
```

However, this requires us to create a new file to hold the result. Sed allows us to modify a file in place using the `-i` switch:

```
$ sed -i.orig email.txt 's/damn/darn/'
```

The `.orig` after the `-i` switch is optional, but useful. If we make a mistake with our sed command, then we could end up losing our original file. If we add the optional extension after `-i`, then it will save a copy of the original

```
1#!/bin/sed -f
2
3s/"\([^\"]*\")"/"\1"/g
4s/'\([^\']*'\)'/'\1'/g
5s/ '\([^\']*'\) ' / '\1' /g
6s/ '\([^\']*'\)$/ '\1'/g
7s/^'\([^\']*'\) '/'\1' /g
8s://""/g
9s://"^[[1m/g
10s://"^[[0m"/g
11s://'^[[1m/g
12s://'^[[0m'/g
```

1 line less; before #3 4 seconds ago
-- INSERT --

12,16-13

file to `email.txt.orig`, just in case we need to restore it again.

Selective printing: a Case Study

The package manager that your author uses to install software on his computer sometimes prints out error messages of the form:

```
...
Installing...
package-name: /path/to/file already exists
in filesystem
package-name: /path/to/file2 already exists
in filesystem
...
Installation failed!
```

This happens when a package tries to install itself by overwriting files that already exist. In this case, the package manager will not overwrite the files, but will pass control back to the user so that they can decide what to do with the file conflicts.

More often than not, it turns out that the files are left over from a broken installation and should be deleted.

“ Sed provides many pieces of functionality besides simple replacement of text ”

In order to delete these packages automatically, it helps to have the names of the offending files printed out line by line. Unfortunately, the package manager prints out a lot of other information besides the paths to the files.

We can use a sed script to print out just the filenames:

```
$ sudo pacman -S package-name 2>&1 | 
  sed -E 's/package-name: ([^ ]*) already
exists in filesystem/\1/' 
...
Installing...
/path/to/file
/path/to/file2
...
Installation failed!
```

This is a good start, but it still prints out input from the package manager that we do not want (for example, the text `Installing...`). We want to print out only those lines that match the pattern, and then apply the substitution.

One approach would be to pipe the output through `grep` before piping it through `sed`:

```
$ sudo pacman -S package-name 2>&1 | 
  grep 'package-name: [^ ]* already exists in
filesystem' |
```

```
sed -E 's/package-name: ([^ ]*) already
exists in filesystem/\1/'
```

But this requires us to type the pattern twice. Luckily, it turns out that we can do the whole thing with one sed command:

```
$ sudo pacman -S package-name 2>&1 |
  sed -E -n 's/package-name: ([^ ]*) already
exists in filesystem/\1/p'
/path/to/file
/path/to/file2
...
```

The option `-n` to Sed suppresses automatic printing. If we use `-n`, then Sed will not print anything out unless we tell it to. The `p` at the end of the `s` command is a command that explicitly tells sed to print out the result after the substitution has been applied. Since we do not apply the substitution to lines that do not match the pattern `package-name: [^]* already exists in filesystem`, then these lines will not be printed out.

We can pipe the output from this pipeline through `xargs rm` in order to delete all of the files.

Address ranges

Besides `s`, there are several other commands that are often used. These commands are typically preceded by an address range, which specifies which lines to apply the command to. An address range is either a single line, a range of lines or a regular expression. To illustrate these, we will use the `p` command which we have already met and the command `seq 10`, which prints out the numbers from 1 to 10.

```
$ seq 10 | sed -n '7p'
7
```

Preceding the command `p` with a single line number means that the command is applied only to that line number. In this case, only the number 7 gets printed out.

```
$ seq 10 | sed -n '4,6p'
4
5
6
```

In the second example, the range `4,6` specifies that we print out lines 4 to 6 of the output.

```
$ seq 10 | sed -n '/../.p'
10
```

This time, we specify that we should only print those lines that match the regular expression `...`. To distinguish it from other commands, the regular expression should be delimited by slashes `/.../`. We can also use regular expressions in range addresses:

```
$ seq 10 | sed -n '1,/[3-5]/p'
```

COMMON SED COMMANDS

#	Treat the command as a comment
q	Immediately quit Sed, printing out the current substitution
d	Delete the current line and move to the next line of input
p	Print out the current line
n	Skip the current line (print it out without applying any substitutions)
{}	Sed treats commands in curly brackets as a group. If we precede a group with an address range, it will be applied to all commands in the group

```
1
2
3
```

Above Sed supports many commands, but the most commonly used (apart from `s`) and most simple are the ones in this table

Here, sed prints out lines from line 1 up till the first line that matches the regular expression `[3-5]`. This time, the third line of output matches that expression, and all subsequent output is suppressed.

Last, we can perform negative matches by appending an exclamation mark `!` after the range address:

```
$ seq 10 | sed -n '2,10!p'
1
```

Here, the command prints out all lines except those in the range 2 to 9.

Note that we can simulate the behaviour of `grep` by writing `sed -n '/regexp/p'` instead of `grep regexp`.

Deleting lines

Besides `s` and `p`, we have access to a number of other useful commands that are available in sed. Like `p`, these can all be preceded by an address range to specify which lines they should be applied to.

We can use the command `d`, for example, to delete a line and move to the next line of input without printing anything. For instance, we can run a sed command to remove the small print at the start of the Moby Dick file that we downloaded in our very first article for this series [Tutorials, p36, [LU&D181](#)]:

```
$ wget https://www.gutenberg.org/files/2701/
old/moby10b.txt >/dev/null
$ sed -i moby10b.txt '1,/END THE SMALL
PRINT/d'
```

This sed command will delete all lines from line 1 up to and including the first line it finds containing the text `END THE SMALL PRINT`.



Adam Oxford

runs South African tech news site www.hxt.co.za. He learned many of these lessons the hard way.

Resources

■ Terminal & editor

All Linux distros have a terminal and most have a number of terminal-based text editors you can access on the CLI

Essential admin commands

20 terminal commands that all Linux web server administrators should know blindfolded

```

669 man df
670 df
671 man df
672 df -h
673 top
674 w
675 nano
676 dir
677 history
678 diff -r /media/studiopc/theRaptorRaid/Pics /media/sdb1/Pics
679 sudo su
680 ssh root@crazyBball.hxt.co.za
681 ssh root@crazyBball.hxt.co.za
682 ssh root@hxt.co.za
683 ssh root@41.79.76.130
684 ifconfig
685 ls | less
686 utst
687 ls | less
688 apt-get update
689 sudo apt-get update
690 history
studiopc@studiopc:~$ █

```

Above Can't remember that really clever thing you did last week? The history command is your friend

Are you an 'accidental admin'? Someone who realised, too late, that they were responsible for the workings of a Linux server and – because something has gone wrong – finds themselves lost in a world of terminals and command lines?

What is SSH?, you may be asking yourself. Do those letters after 'tar' actually mean anything real? How do I apply security patches to my server? Don't worry, you're not alone. And to help you out, we've put together this quick guide with essential Linux commands that every accidental admin should know.

Becoming an accidental admin

While we'd argue that they should, not everyone who starts using Linux as an operating system does so through choice. We suspect that most people's first interaction with Linux happens somewhat unwittingly. You click a button on your ISP's account page to set up a personal or business web server – for a website, email address or online application – and suddenly you're a Linux admin. Even though you don't know it yet.

When starting out with your web server, things are usually straightforward. Nearly all hosting providers give you a web interface such as cPanel or Plesk to manage your server. These powerful pieces of software give quick and easy access to logs, mail services and one-click installations of popular applications such as WordPress. But the first time you have to do something that isn't straightforward to do through the graphical control panel, you're suddenly out of the world of icons and explanatory tooltips and into the world of the text-only terminal.

To make things worse, for a lot of people the first time they have to deal with the terminal is when something has gone wrong and can't be fixed through the control panel. Or perhaps you've just read that there's a major

security flaw sweeping the web and all Linux servers must be updated at once. Suddenly you realise that your nice control panel hasn't actually been updating your server's operating system with security patches and your small personal blog may well be part of a massive international botnet used to launch DDOS attacks against others. Not only are you a stranger in a strange land, you're probably trying to recover or fix something that was really important to you, but which you never gave much thought to while it was being hosted for a couple of pounds a month and seemed hassle-free.

You are an 'accidental admin'. Someone who is responsible for keeping a Linux web server running and secure, but you didn't even realise it. You thought all that was included in the couple of pounds a month you pay to your ISP – and only found out it's not when it was too late.

Since most web servers are running Ubuntu, this guide is based on that particular distribution. And all the commands here are just as applicable to a Linux desktop as they are to a web server, of course.

01 sudo

The most fundamental thing to know about Linux's approach to administration is that there are two types of accounts that can be logged in: a regular user or an administrator (aka 'superuser'). Regular users aren't allowed to make changes to files or directories that they don't own – and in particular this applies to the core operating system files, which are owned by an admin called 'root'.

Root or admin privileges can be temporarily granted to a regular user by typing **sudo** in front of any Linux command. So to edit the configuration file that controls which disks are mounted using the text editor, nano, you might type **sudo nano /etc/fstab** (we really

Above Even if someone copies your key, they'll still need a password to unlock it

“ Root or admin privileges can be temporarily granted to a regular user by typing `sudo` in front of any Linux command ”

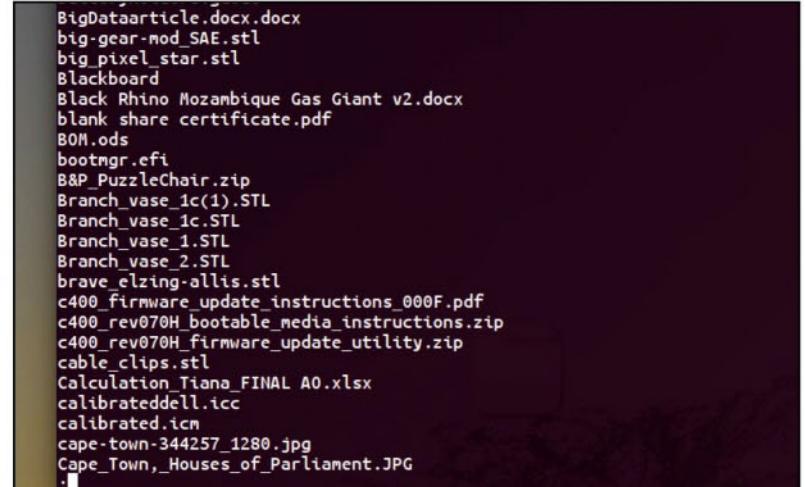
don't recommend this unless you know what you're doing). After entering `sudo`, you'll be asked for your user password. On a desktop PC, this is the same one that you use to log in. If you're logging into your own web server, however, there's a good chance that you'll already be the root user and won't need a password to make important changes.

If you can't execute `sudo` commands, your web host has restricted your level of access and it probably can't be changed. User accounts can be part of 'groups' in Linux and only members of the sudoers groups can use the `sudo` command to temporarily grant themselves admin privileges.

02 **su** While `sudo` gives you great power, it still has limitations. Most of all, if you've got a whole bunch of commands to enter, you don't want to have to type it out at the start of every single line [at least the password has a 5 minute timeout – Ed]. This is where `su` comes in, which will give you superuser powers until you close the terminal window. Type `sudo su` followed by your password, and you'll see the prompt change from `yourname@yourserver` to `root@yourserver`. You might think `su` stands for superuser, but it's actually a command to change to any user on the system and if it's used without an account name after it, `su` assumes you want to be root. However, using `su myname` will switch you back to your original, non-super, login.

03 **ifconfig** Since you're troubleshooting a web server, it's probably a good idea to get as many details about its actual connection as possible noted down. The `ifconfig` command can be run without `sudo` privileges and tells you details about every live network connection, physical or virtual. Often this is just for checking your IP address, which it reports under the name of the adaptor, but it's also useful to see if you're connected to a VPN or not. If a connection is described as `eth0`, for example, it's an Ethernet cable; meanwhile, `tun0` is a VPN tunnel.

04 **chown** There's tons more you can learn about `chmod` (see box, p33) and we strongly recommend that you do, but it has a sister command that's even more powerful. While `chmod` dictates what users who aren't the owner of a file can do, the `chown` command changes the file owner and group that it belongs to completely. Again,



A screenshot of a terminal window showing a list of files in a folder. The files listed include: BigDataarticle.docx.docx, big-gear-mod_SAE.stl, big_pixel_star.stl, Blackboard, Black Rhino Mozambique Gas Giant v2.docx, blank share certificate.pdf, BOM.ods, bootmgr.efi, B&P_PuzzleChair.zip, Branch_vase_1c(1).STL, Branch_vase_1c.STL, Branch_vase_1.STL, Branch_vase_2.STL, brave_elzing-allis.stl, c400_firmware_update_instructions_000F.pdf, c400_rev070H_bootable_media_instructions.zip, c400_rev070H_firmware_update_utility.zip, cable_clips.stl, Calculation_Tiana_FINAL A0.xlsx, calibrated.dll.icc, calibrated.icm, cape-town-344257_1280.jpg, Cape_Town,_Houses_of_Parliament.JPG, and a file named with a question mark. The terminal prompt is visible at the bottom.

you'll probably need to put `sudo` in front of anything you `chown`, but the syntax is simple. An example might be `sudo chown myname:mygroup filename.file`.

Above Unless you can read 1,000 lines a second, you'll need to use `ls` | `less` to explore folders

05 service restart

No, we're not telling you to 'try turning it off and on again', but sometimes it's a good place to start (and sometimes it's essential to load changes into memory). It's possible you might be used to start and stop background processes on a Windows desktop through the graphical System Monitor or Task Manager in Windows. However, in the command-line terminal to a server it's a little more tricky, but not by much.

Confusingly, because many Linux distributions have changed the way they manage startup services (by switching to `systemd`), there are two ways of doing this. The old way, which still works a lot of the time, is to just type `service myservice restart`, preceded with `sudo` when it's necessary. The new, correct, way is a little more verbose: `systemctl restart myservice.service`. So if you want to restart Apache, for example, the core software which turns a mere computer into a web server, the command required would be `sudo systemctl restart apache2.service`.

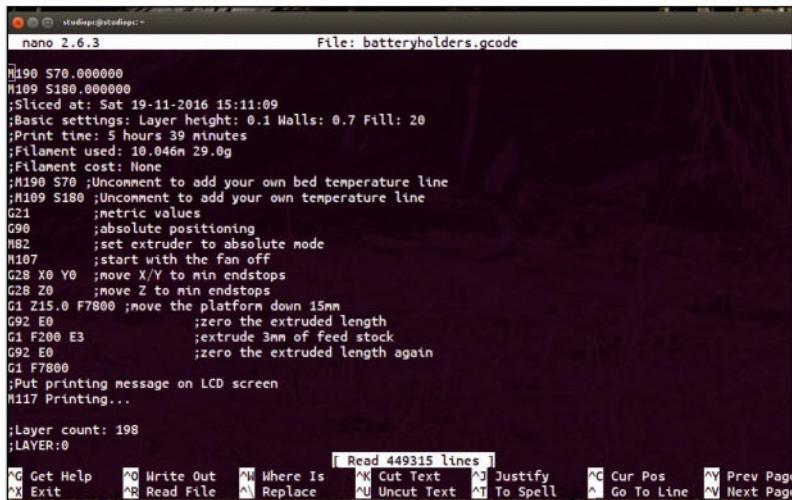
06 ls

The key to understanding the console is all in the path, shown to the left of the command prompt, which tells you whereabouts you are in the folder structure at any given time. But how do you know what else is in your current location? Easy: you use `ls`. The `ls` command lists all the files within the folder that you're currently browsing. If there's a lot of files to list, use `ls` | `less` to pause at the end of each page of filenames.

07 cat

A command you'll often see if you're following instructions you've found online – and aren't always sure what you're doing – `cat` is short for 'concatenate' and is used to combine files together. In its simplest form it can be used to take `file1.txt` and `file2.txt` and turn them

■ Recursive
If you are changing names, permissions or ownership, most commands have a `-R` or `-r` option, which stands for 'recursive'. Essentially, this changes the attributes of all files inside a folder, rather than just the folder itself.



```

M100 S70.00000
M109 S180.00000
;Sliced at: Sat 19-11-2016 15:11:09
;Basic settings: Layer height: 0.1 Walls: 0.7 Fill: 20
;Print time: 5 hours 39 minutes
;Filament used: 10.046m 29.8g
;Filament cost: None
;M109 S70 ;Uncomment to add your own bed temperature line
;M109 S180 ;Uncomment to add your own temperature line
G21 ;metric values
G90 ;absolute positioning
M82 ;set extruder to absolute mode
M107 ;start with the fan off
G28 X0 Y0 ;move X/Y to min endstops
G28 Z0 ;move Z to min endstops
G1 Z15.0 F7800 ;move the platform down 15mm
G92 E0 ;zero the extruded length
G1 F200 E3 ;extrude 3mm of feed stock
G92 E0 ;zero the extruded length again
G1 F7800
;Put printing message on LCD screen
M117 Printing...
;Layer count: 198
;LAYER:0

```

Get Help Write Out Where Is Cut Text Justify Cur Pos Prev Page
 Exit Read File Replace Uncut Text To Spell Go To Line Next Page

Above Nano isn't the only terminal text editor, but it's the easiest to use

into file3.txt, but it can also be combined with other commands to create a new file based on searching for patterns or words in the original.

Quite often you'll see **cat** used simply to explore a single file – if you don't specify an output filename, **cat** just writes what it finds to the screen. So online walkthroughs often use **cat** as a way of searching for text within a file and displaying the results in the terminal. This is because **cat** is non-destructive – it's very hard to accidentally use **cat** to change the original file, whereas other commands might do.

08 find

A useful and underused command, **find** is pretty self-explanatory. It can be used to find stuff. Typing it by itself is much like **ls**, except that it lists all the files within subdirectories of your current location as well as those in your current directory. You can use it to search for filenames using the format **find -name "filename.txt"**. By inserting a path before the **-name** option, you can point it at specific starting folders to speed things up. By changing the **-name** option you can search by days since last accessed (**-atime**) or more.

09 df

Maybe your server problems are to do with disk space? Type **df** and you'll get a full breakdown of the size and usage of every volume currently mounted on your system. By default it'll give you big numbers in bytes, but if you run **df -h** (which stands for 'human readable'), the volume sizes will be reported in megabytes, gigabytes or whatever is appropriate.

10 apt-get update && upgrade

Probably the single most important command to know and fear. We all know that to keep a computer system secure you need to keep it updated, but if you've got control of a Linux box the chances are that it isn't doing that automatically.

A simple **sudo apt-get update** will order your system to check for the latest versions of any applications it's

running, and **sudo apt-get upgrade** will download and install them. For the most part these are safe commands to use and should be run regularly – but occasionally updating one piece of software can break another, so make a backup first...

11 grep

As computer commands go, there are few more fantastically named for the newcomer than **grep** [it's a real verb! – Ed]. How on earth are you ever going to master this Linux stuff if it just makes words up? But **grep** is a great utility for looking for patterns within files. Want to find every line that talks about Cheddar in a book about cheeses? **grep "cheddar" bookofcheese.txt** will do it for you. Even better, you can use it to search within multiple files using wildcards. So **grep "cheddar" *.txt** will find every text file in which Cheddar is referenced. So now you grok **grep**, right?

12 top

When you're working in a graphical user interface such as a Linux desktop environment or Windows desktop, there's always an application like System Monitor or Task Manager which will call up a list of running applications and give you details about how many CPU cycles, memory or storage they're using. It's a vital troubleshooting tool if you have a program that's misbehaving and you don't know what it is.

In a similar way, you can bring up a table of running applications in the Linux terminal that does the same thing by typing **top**.

Like a lot of command-line utilities, it's not immediately obvious how you can close **top** once you're finished with it without closing the terminal window itself – the almost universal command to get back to a prompt is **Ctrl+C**.

13 kill, killall

Using **top**, you can figure out which application is using all your CPU cycles, but how do you stop it without a right-click > End process menu? You use the command **kill** followed by the process name. If you want to be sure and kill every process with a name that contains that application name, you use **killall**. So **kill firefox** will close down a web browser on a Linux desktop.

14 w

From the weirdness of **grep** to the elegance of the **w** command, a whole command in a single letter. If you think another user is logged into your system, this is an important command to know. You can use **w** to list all currently active users, although don't rely on it too much as it's not hard for a hacker to stay hidden.

"To be sure and kill every process with a name that contains that application name, you use killall "

Above Keep an eye on the directory path in front of the command line to figure out where you are

15 passwd

15 You must use `passwd` with extreme care. Ultra extreme care. Because the next word you write after it will become your login password, so if you type it incorrectly or forget it, you're going to find yourself in serious trouble.

You can only change your own user's password by default, but if you grant yourself sudo powers you can change any user's credentials by including their username after the password itself. Typing `sudo passwd`, meanwhile, will change the password for root.

“ If you grant yourself sudo powers, you can change any user’s credentials ”

Check out the manual (`man passwd`) page for some useful options to expire passwords after a certain period of time and so on.

16 cd

18 If you have a graphical interface and file browser, it's pretty easy to move to new locations on your hard drive just by clicking on them. In the terminal, we know where we are because of the path (shown to the left of the command prompt), and we switch location using `cd`, which stands for 'change directory'.

The `cd` command is mainly used in three ways:

- `cd foldername` – This will move you to that folder, provided it exists within the folder you're currently browsing (use `ls` if you're not sure).
 - `cd ~/path/to/folder` – This will take you to a specific location within your **home** folder (the `~` character tells `cd` to start looking in your **home** folder). Starting with a `/` will tell `cd` to start the path at the **root** folder of your hard drive.
 - `cd ..` – This final useful command simply takes you up one level in the folder structure.

17 mv & rm & cp

When you get the hang of it, using a terminal as a file manager becomes pretty simple and quite a joyful experience. As well as `cd`, the three fundamental commands are `mv`, `rm` and `cp`. The `mv` command is used to move a file from one location to another, `rm` is

used to remove or delete a file, while `cp` will copy files and folders.

Just as with the `cd` command, you can either enter a filename to operate on a file in the directory you're working in or a full path starting from the root of the drive with `~`. For `mv`, the syntax is `mv ~/location1/file1.file ~/location2/location`.

The big thing to remember is that in the terminal there's no undo or undelete function: if you `rm` a file it's gone forever (or at least will require very specialist skills to retrieve) and in a similar fashion, if you `mv` or `cp` a file you'd better make a note of where it went.

18 nano

It might seem odd, if you've spent your life in graphical applications and utilities, but complex programs run in the text terminal, too. There are several text editors which normally come as part of the whole package, notably nano and vi. You can open a blank document by typing `nano`, or you can edit an existing one by typing `nano ~path/to/text.txt` (and do the same with vi). Some of the terminology may seem odd, though: To write out (`Ctrl+O`) means save, for example and so on.

19 history

If you've been copying and pasting commands from the web all day, you might want to check what you've actually done. You can use `history` to give you a list of all the terminal commands entered going back a long, long way. Execute specific numbered commands with `!<num>`. You can go back through recent commands just by using the up and down arrows (and reissue them by tapping `Enter`), or search for commands by pressing `Ctrl+R`.

■ 20 chmod

A file which can be seen by web visitors, but can only be changed by a specific user, is just about as basic as it gets when it comes to locking down a server. The problem is that some files need to be changeable and some don't – think of a WordPress installation for a blog. You want WordPress to be able to write some files so it can update them, but there's also a lot of files you don't want it to be able to change – and you really don't want to give it power to execute code unless you have to. The flip side is that problems with web servers can be traced back to incorrect file permissions, when an app needs to be able to modify a file but has been locked out by default.

■ Man up

One command that's invaluable is `man`, which is short for 'manual'. This will open up the help file for any other command. So if you want to know all the options for the `ls` command, simply type `man ls` and see what comes up.



Tam Hanna

is the CEO of the Bratislava-based consulting company Tamoggemon Holding k.s. The firm's focus is consulting in the development of interdisciplinary systems consisting of software, HID and hardware. Tam often makes screen recordings for his work.

Resources

SimpleScreen Recorder

<http://bit.ly/SSRecorder>

Configuration for YouTube recordings

<http://bit.ly/SSR-YouTube>

Paper analysing frame rates used in e-learning scenarios

<http://bit.ly/Screencast-elearning>

SimpleScreenRecorder

Video capture on Linux

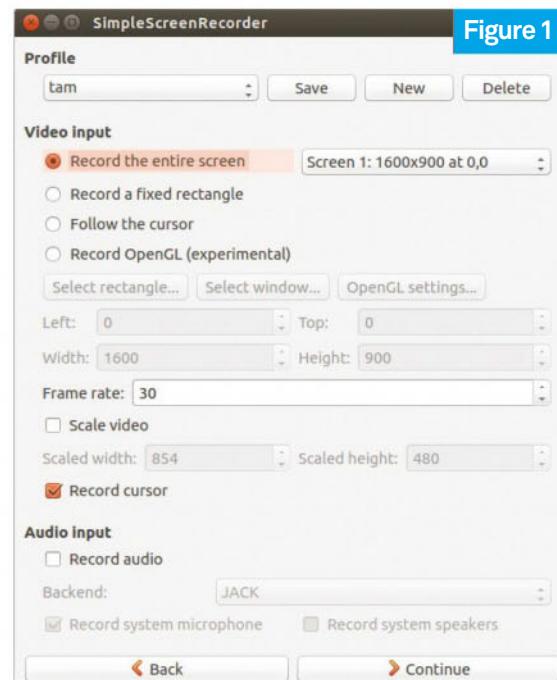
Learn to use SimpleScreenRecorder for training purposes and make how-to videos worth their file size in gold

The old-school approach to recording a walkthrough used to involve pointing a camera at your screen – and you still see those on YouTube, especially if someone is covering BIOS/EFI guidance. While this works in theory, it is grossly inefficient: keep in mind that your computer generates digital picture information, which is then output onto an analogue medium. An analogue camcorder then re-digitsises this information, only to store it into a video file. In addition to the effort involved in getting the alignment right, digitisation and quantisation noise never sleep.

Capturing programs used to be difficult because of the high processing power required for encoding live videos coming from a framebuffer. Fortunately, the development of ever-faster processors ameliorated this problem. For instance, your author often records live video of Android Studio and similarly demanding products such as PCB design software, IBF Target3001 on a five-year-old IBM ThinkPad T430 with a relatively slow dual-core Intel CPU.

A question of tooling

Looking for screen-recording solutions in your repository of choice will yield a huge number of options. However, selecting a reliable one is not easy. For instance, we've experimented with Kazam and this turned out to be a disaster, as the program regularly created corrupted



Above The first step is all about selecting the right data source



Figure 2

Above Manage your multiple screen setup carefully, in this case, two screens are next to one another and have a slight offset

footage. SimpleScreenRecorder by Maarten Beert can be considered the gold standard and it's used by both YouTubers and educators. The following steps to install it assume that you're using Ubuntu 14.04:

```
sudo add-apt-repository ppa:maarten-beert/
simplescreenrecorder
sudo apt-get update
sudo apt-get install simplescreenrecorder
```

Users on a more recent version of the operating system can download the program directly, as Canonical added the product to the repositories for 17.04:

```
sudo apt-get update
sudo apt-get install simplescreenrecorder
```

If you use a 32-bit 3D application on a 64-bit workstation, you will need to install an additional module for recording your output:

```
sudo apt-get install simplescreenrecorder-
lib:i386
```

SimpleScreenRecorder isn't limited to Ubuntu Linux. The program can be used on almost any *NIX operating system. For more information on support head to <http://bit.ly/SSRecorder>.

With that out of the way, the program can be started. Simply type its name to open the main screen (shown in **Figure 1**). When working with SimpleScreenRecorder, start at the top and look downward. The first toggle allows you to select the source of the material that you want to record. One very important aspect relates to systems which have more than one screen attached: the combo box next to 'Record the entire screen' provides a variety of options. Individual screens are identified

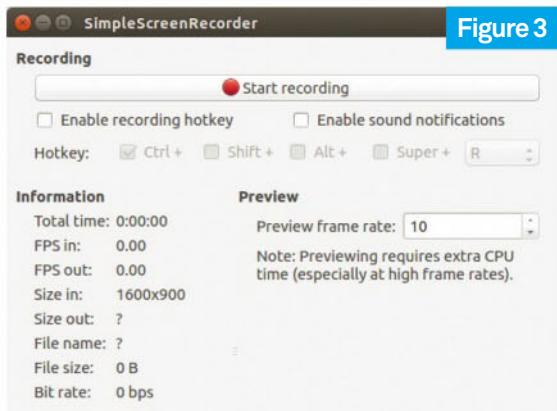


Figure 3

Above Once you've chosen your recording settings, you get a dialog that's ready and waiting for you to press 'start recording'

by the number assigned to them by Ubuntu's display stack: if you want to record one display, you simply select its name.

The selection for all screens must be managed with care. When working with multiple screens, most distros – Ubuntu included – try to map the display in a fashion similar to the way they are arranged on the screen. This slightly confusing concept is illustrated in **Figure 2**.

When ordered to record all screens, SimpleScreenRecorder grabs the entire bitmap out of the framebuffer. If it contains a large amount of garbage or unwanted black space, simply rearrange the display organisation in display settings before starting the actual recording procedure.

Keeping the amount of information recorded in check can be a challenge. SimpleScreenRecorder provides a number of recording options, including 'Record a fixed rectangle' and 'Follow the cursor'. The former requires you to designate an area on the screen, whose contents are then recorded. Ergonomic experience speaks against using the second mode of operation – the large amount of motion tends to cause nausea in viewers in a relatively short time.

Two options remain on the screen: first of all, you can enable the record cursor option if you want your mouse pointer to show up during the recording. This can be beneficial when recording click-by-click instructions: if the cursor is missing, viewers have to concentrate harder and will become tired more quickly. Furthermore, the frame rate and the audio recorder can be enabled. Given that audio tends to be the biggest problem for aspiring video creators, be careful with this – the microphones of most PCs, while perfectly sufficient for VoIP, offer awful quality for recordings.

With the initial recording options out of the way, click the 'Continue' button to switch to the second part of the setup assistant. SimpleScreenRecorder provides a large number of options for tailoring the output. We usually select an MP4 container and use the H.264 codec for encoding. Generally, set the fastest possible encoding options: using a bit more memory (in the range of less than a megabyte per second) is usually a good

trade-off against using up more processor resources which are better spent on the program you are trying to demonstrate. Also, make sure to select the checkbox 'Allow frame skipping'. It is worth its weight in gold should your computer's processor be overloaded during the recording process. Finally, make sure to set up the file storage properties correctly – accidentally overwriting production-ready footage can be catastrophic.

Going live

SimpleScreenRecorder's GUI contains another gotcha: clicking 'Continue' opens a screen (**Figure 3**). This does not enable recording – the actual recording process starts only after the 'Start recording' button is clicked.

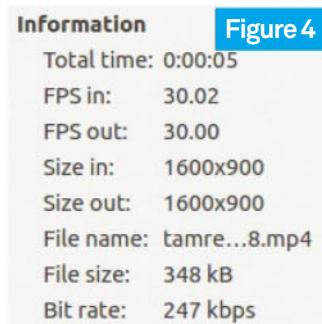
With that sorted, the actual recording can commence. The moment 'Start recording' is clicked, the program starts collecting information. Counters displayed on the screen provide an overview of the current status (**Figure 4**) – if a significant imbalance (more than 1fps or so) is shown between the two values, the CPU is overloaded.

One very helpful feature of Simple ScreenRecorder is its native support for breaking apart footage. If you click 'Pause recording', the program will commit the currently recorded information to the disk. Pressing 'Start recording' once again leads to the creation of a new recording. In addition to that, the log field at the bottom of the screen provides an overview of recent events. Finally, click 'Save recording' to complete the recording process and commit the final bit of information to your disk.

Should you ever feel paranoid about the contents of the recording, click the 'Start preview' option. SimpleScreenRecorder will react to this by showing a small on-screen window with recently captured information: be aware that it updates much slower than the actual information, to save CPU performance.

The profile manager displayed at the top of most SimpleScreenRecorder dialogs enables you to create and restore profiles. With them, settings can be stored and recalled comfortably: this is very helpful in that it allows you to quickly reload commonly used settings. The developer, furthermore, provides a set of predefined settings for some commonly used recording scenarios.

Creating good tutorial videos is an art in itself: recording perfect screencasts is but part of the problem. The best footage will not go far if it presented confusingly and unstructured way. However, obtaining great – and noise-free – video will simplify getting the rest right, as didactic skills can improve as you practise and audio quality can be improved with the purchase of a better microphone, such as Blue Yeti. ■



Left The number of frames going in and out should be balanced

■ A question of frame rate Gamers show a knee-jerk preference for high frame rates. However, when working on a tutorial, a YouTube show or something similar, needlessly cranking up the frame rate is unwise. A study performed by Springer shows that more than 50 per cent of non-game screencasts are at 15fps (see <http://bit.ly/ScreenCast-eLearning>).



Tam Hanna

is the CEO of consulting company Tamoggemon Holding k.s. He grew up under the influence of Eric Sink's classic essays on coupling and event oriented-programming.

Resources

<https://mosquitto.org>

The website of the Mosquitto server used to host our infrastructure

<http://mqtt.org>

The website of the MQTT protocol community

www.hivemq.com

A professional, albeit pricey, alternative to Mosquitto

PART ONE

MQTT: An intro to the protocol powering IoT

Use MQTT and set up a Mosquitto server to send data between resource-constrained embedded devices

Event-oriented architectures are ideally suited to many tasks. For example, take a system that forwards humidity sensor data to various clients. The biggest benefit is that the arrangement between producers and consumers can be organised on the fly: if readings must also be forwarded to a logger, no changes in the creator (the sensor) or the other consumers are required.

Furthermore, testing event-driven architectures is greatly simplified compared to traditional systems. This phenomenon is explicable if you look at the way in which parts are tied together – replacing the aforementioned humidity sensor with a mock-up class is really simple: you implement the event-sourcing protocol and proceed to sending prepared data to the clients.

Event-based systems are nothing new. Pivotal in the world of IoT, MQTT is a lightweight messaging protocol for small sensors and mobile devices, optimised for high-latency or unreliable networks. Its main benefit is that it is an industry standard: developers working on a custom

protocol must implement the network code on every single platform; MQTT users simply download the library. Furthermore, developers are likely to already know how to handle MQTT: this simplifies cooperative scenarios in smart-home and similar high-value environments. Finally, there is a large variety of commercial offerings such as hosted MQTT brokers available, which means that scaling problems lose a lot of their bite.

Most MQTT tutorials start with a detailed description of the protocol, only to look at practical applications afterwards. As this approach can be tedious (and we have a three-part series), let's take a different approach. We'll start out by setting up the Mosquitto server and will then connect all kinds of add-ons to it.

Don't install the Mosquitto server on a virtual machine: the next part of the series will deploy advanced hardware which must be on the same network as the MQTT server.

Fortunately, the open source Mosquitto is available in most, if not all, package repositories. When working



Above Fun fact: MQTT was originally designed in 1999 by Andy Stanford-Clark and Arlen Nipper for monitoring oil and gas pipelines

on Ubuntu 14.04, the product can be deployed via `sudo apt-get install mosquitto`.

The following steps assume that the Mosquito server is on a private network: securing it for use on a public server isn't covered in this series. Mosquitto implements a publish-subscribe event system. This means that clients must announce their interest in a specific channel, and will, from then on, be provided with all events matching the string. Let's start out by demonstrating this feature with the Mosquitto command-line clients, which are not part of the distribution. They can be downloaded by entering `sudo apt-get install mosquitto-clients`.

MQTT's channels are generated dynamically. Open a new terminal on your development workstation and enter `mosquitto_sub -h localhost -t tamstest`. The Mosquitto subscription agent will remain blocked until messages come in: the `-t` parameter specifies which channel is to be observed. The distribution comes with an additional helper responsible for creating MQTT events:

```
mosquitto_pub -h localhost -t tamstest -m "hello world"
```

The moment you press the **Return** key in the terminal containing the `mosquitto_pub`, the first terminal will show a 'hello world' message (similar to **Figure 1**).

Bring in the Qt

The MQTT module for the C++ framework is not complete as of version 5.10. Due to that, we must make do with a beta version from GitHub (<https://github.com/qt/qtmqtt>).

As integrating the provided module into a local Qt built gets tedious, **LU&D** provides you with a ready-to-go project skeleton. It is based on beta 3 of Qt 5.10, and is included with the code accompanying this tutorial. It was created by copying the relevant implementation files into the project, and adjusting some of the includes to the new module-less situation.

Load it and open the header for the `MainWindow`. Then, make sure to include a header and to create a pointer for the MQTT client class instance which we will use during the following steps:

```
#include "mqtt/qmqttclient.h"
. . .
class MainWindow : public QMainWindow . . .
private:
    Ui::MainWindow *ui;
    QMqttClient *m_client;
};
```

With that done, let's proceed to connecting to a server in the constructor of `MainWindow`:

```
MainWindow::MainWindow(QWidget *parent) : . .
{
    ui->setupUi(this);
    m_client = new QMqttClient(this);
```

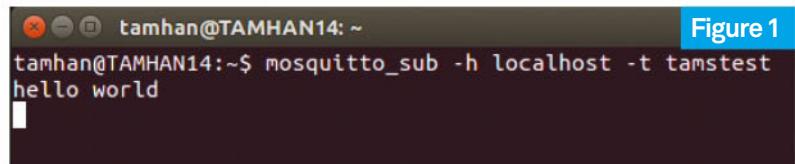


Figure 1

```
m_client->setHostname("test.mosquitto.
org");
m_client->setPort(1883);
connect(m_client,
&QMqttClient::stateChanged, this, &MainWindow::up
dateLogStateChange);
connect(m_client,
&QMqttClient::disconnected, this, &MainWindow::br
okerDisconnected);
m_client->connectToHost();
}
```

Above Here `mosquitto_sub` waits for incoming messages and does not return until terminated

After creating a new instance of the `QMqttClient` class, we set the address of the Mosquitto test server: the developer team behind the server provide a reference implementation so that developers can perform quick integration tests. Port number 1883 is considered standard for unencrypted MQTT – enabling encryption features is not covered in this guide.

“ The MQTT module for the C++ framework is not complete as of v5.10 – we must make do with a beta version from GitHub ”

Finally, two event listeners are registered so that we can collect information about the connection progress. They output messages into the `qDebug` log – `updateLogStateChange` looks like this:

```
void MainWindow::updateLogStateChange() {
    const QString content =
QDateTime::currentDateTime().toString()
        + QLatin1String(":"
State Change")
        + QString::number(m_
client->state());
    qDebug() << content;
}
```

Connection status information is given in the form of a enum – be aware that a successful connection leads to the value 2 being stored in the `state` field.

With that accomplished, it's time to test our program for the first time. The output of `StateChange` will inform us that the test service has accepted our connection – in theory, we can now use our own server

Beware of licensing!

As of this writing, the Qt company has not completed the licensing procedures on QtMqtt. Be careful when including the code in a commercial application which is not GPL licensed and expect – usually minor – API changes as the product achieves technical maturity.

■ Dynamic logging

Mosquitto is not limited to outputting debugging information to the syslog. It can also make status information available via a topic, which random MQTT clients can register in order to stay in the loop. Further information on this – very advanced – use case can be found by visiting <http://bit.ly/SYSTopics>.

by adjusting the connection parameters passed into the `MQTTClient` object:

```
MainWindow::MainWindow(QWidget *parent) :
    ...
    m_client->setHostname("localhost");
    m_client->setPort(1883);
```

Sadly, running this modified version is unsatisfactory. After outputting 'State Change 1' to tell us the connection process is started, a broker-lost message is emitted. The server is actively refusing our connection requests.

Hunting down trouble

Mosquitto is supported by a wide array of logging facilities. Sadly, most of them are not enabled by default. Firing them up requires editing of the file `/etc/mosquitto/mosquitto.conf` – pick an editor of choice, start it with superuser rights and get to work. All important parameters are found in the segment logging – the logging destination must be set up by passing in the string `syslog`:

```
log_dest syslog
```

Next, look for the following lines. They are prefixed with a `#` character by default, which marks them as disabled. Simply remove the character to enable the relevant logging features and then save the file:

```
log_type error
log_type warning
log_type notice
log_type information
connection_messages true
```

Restart the Mosquitto server to force it to read the new configuration file by entering `sudo service mosquitto restart`. Looking at logging info contained in the system log is best accomplished using the syslog viewer – enter 'System Log' and feast your eyes on the contents. Then, reconnect your client to get the error shown in **Figure 2**.

This problem is caused by a versioning mismatch between Ubuntu and QtMQTT's implementations: the

Below Our Qt program sends something which our server does not accept

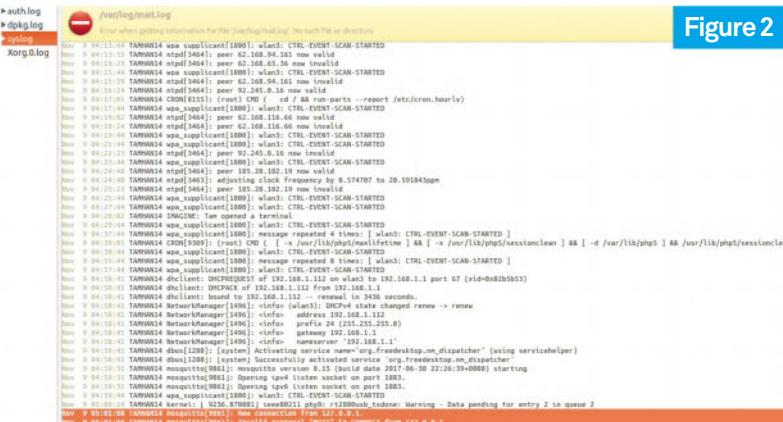


Figure 2

MQTT protocol received an update which is implemented in both QtMqtt and the official Mosquitto test server. However, the team behind the Ubuntu package manager is known for being slow to update things. Due to that, a bit of manual intervention is needed.

First of all, return to the syslog – a starting instance of Mosquitto will announce its version. Be careful to ignore the build date – it is relevant only in that it describes when Ubuntu's build server last touched the code, and has nothing to do with the freshness of the code base used.

Next, remove the existing version of both client and server from your workstation by entering `sudo apt-get remove mosquitto mosquitto-clients`. Finally, add a new PPA to apt-get's list, perform update and reinstall Mosquitto:

```
sudo add-apt-repository ppa:mosquitto-dev/
mosquitto-ppa
sudo apt-get update
sudo apt-get install mosquitto
```

During the deployment of the new package, a message will pop up. Apt-get asks if you want to use the current configuration file or prefer to replace it with the default one provided in the new version of the package.

As we want to keep the logging commands intact, select the **N** option and press **Enter** to keep the existing configuration file. After that, that system log file will show a new version of the server start up automatically: its build date might be older than the one obtained from the official repository; its version number will, however, be in the range of 1.4.x, indicating a much newer version of the underlying code base (see **Figure 3**).

A look at channel matching

Let us modify our Qt program so that it registers itself for our test message. First of all, the creation of additional fields in the header of the main form is required:

```
#include "mqqt/qmqttsubscription.h"
class MainWindow : public QMainWindow
{
    ...
    QMqttClient *m_client;
    QMqttSubscription *mySub1, *mySub2,
*mySub3;
};
```

QtMQTT describes subscription relationships via the `QMqttSubscription` class. We create a total of three instances here – this makes adding further subscriptions easier when you want to handle more than one of them.

At this point, a significant question remains: where do we start the actual subscription process? The answer to this can be found in `updateLogStateChange`:

```
void MainWindow::updateLogStateChange() {
    ...
    if(m_client->state() == QMqttClient::ClientState::Connected) {
```

```

        mySub1=m_client-
>subscribe("tamstest",0);
        connect(mySub1, &QMqttSubscription::messageReceived, this,&MainWindow::updateMessage);
        connect(mySub1, &QMqttSubscription::stateChanged, this, &MainWindow::updateStatus);
    }
}

```

After outputting information into the debug console, the event handler checks the connection state of the MQTT client. If a connection to the broker has been established successfully, the `subscribe()` method is invoked in order to spawn a subscription class. Its constructor requires both a string describing the channel and a numeric QoS field – the letter will be discussed in next issue's tutorial.

We finally install two signal handlers in order to make sure that our program is notified of significant events:

```

void MainWindow::updateMessage(const
QMqttMessage &msg) {
    qDebug() << "Got " << msg.payload();
}
void MainWindow::updateStatus(QMqttSubscription::SubscriptionState state) {
    qDebug("State Change!");
}

```

With that done, restart the program. Enter the `mosquitto_pub -h localhost -t tamstest -m "hello world"` command to fire off another message. It will show up in the debug log.

Complex games!

Using strings to correlate between event types works as long as the number of events is small. One of the design goals of MQTT is the ability to handle a very large number of events. For this, some kind of structure is required: for example, feeds could be structured to have parents and children.

This is accomplished by the creation of a folder hierarchy: the '/' character is used to keep the individual elements apart. Let us look at the following structure, which could represent a network of devices inside of two laboratories owned by Tamoggemon Group:

```

tamoggemon/labbratislava/dpo1
tamoggemon/labbratislava/lecroy1
tamoggemon/labvac/lecroy1
tamoggemon/labvac/lecroy2

```

From Qt's point of view, subscribing to such an event is not particularly difficult. Let's subscribe to events from the first LeCroy oscilloscope in Vac – the main issue is that the strings are case sensitive:

```

if(m_client->state()==QMqttClient::ClientsState::Connected)
{
    mySub1=m_client-

```

Figure 3

```

Nov 11 02:23:02 TAMHAN14 mosquitto[21769]: mosquitto version 1.4.12 (build date Sat, 27 May 2017 21:38:19 +0100) sta-
Nov 11 02:23:02 TAMHAN14 mosquitto[21769]: Config loaded from /etc/mosquitto/mosquitto.conf.
Nov 11 02:23:02 TAMHAN14 mosquitto[21769]: Opening ipv4 listen socket on port 1883.
Nov 11 02:23:02 TAMHAN14 mosquitto[21769]: Opening ipv6 listen socket on port 1883.

```

```

>subscribe("tamoggemon/labvac/lecroy1",0);
        connect(mySub1, &QMqttSubscription::messageReceived, this,
&MainWindow::updateMessage);

```

Above Mosquitto 1.4 fully implements more recent MQTT standards

Dispatching an event simulating this unlucky oscilloscope is accomplished via `mosquitto_pub -h localhost -t tamoggemon/labvac/lecroy1 -m "hello world"`. The interesting aspect of topics is the use of wildcards: it allows programs to sign up for a group of events with a single operation. We can, for example, subscribe to all events emanating from all devices based in Vac:

```

if(m_client->state()==QMqttClient::ClientsState::Connected)
{
    mySub1=m_client-
>subscribe("tamoggemon/labvac/#",0);

```

“ One of the design goals of MQTT is the ability to handle a very large number of events ”

Alternatively, we can disable one layer of the hierarchy. For example, let's assume that you are interested in all the events generated by an oscilloscope that we've called lecroy1:

```

if(m_client->state()==QMqttClient::ClientsState::Connected)
{
    mySub1=m_client-
>subscribe("tamoggemon/+/lecroy1",0);

```

While this is quite flexible, keep in mind that MQTT does not allow for complex string matching. For example, it is not allowed to use matchers such as `lecroy*` to announce interest in all LeCroy units.

Furthermore, changing the structure of an MQTT deployment tends to be extremely difficult: due to this, developers are very advised to think carefully about how they structure their product before the first piece of hardware gets sent out into the field.

Running MQTT on the desktop is child's play: packet loss is unlikely. This is not the case in the real world. MQTT contains a variety of features intended to mitigate or alleviate the consequences of connection losses. In the next issue, we will deploy MQTT on Android Things. Until then, may your packages always arrive safely. ■



Toni Castillo Girona

holds a degree in Software Engineering and an MSc in Computer Security and works as an ICT research support expert in a public university in Catalonia (Spain). Read his blog at <http://disbauxes.upc.es>.

Resources

Burp Suite Free Edition

<http://bit.ly/BurpSuite>

Wireshark

`apt-get install wireshark`

Arpspoof from the Dsniff package

`apt-get install dsniff`

NoPE Proxy

<http://bit.ly/NopeProxy>

A GNU/Linux computer

An Android device

Redirect and intercept HTTP requests from devices

By redirecting traffic from phones and IoT devices, you can detect malicious behaviour, find weak spots and more

Proxies are great tools for pen-testing web applications. They are capable of detecting common vulnerabilities on websites (SQLi, XSS, sensitive data exposure, and so on), while providing you with a powerful framework for circumventing website client-controls, performing manually scans or conducting automated attacks. A lot has been written about using either Burp or ZAP! proxies during a web pen-testing engagement. But it is not all about the web – think about all those embedded and IoT devices making use of the HTTP protocol behind the scenes in order to send or receive data, communicate with some server in the cloud, or just perform some other unattended tasks. Not to mention smartphones running either Android or iOS! Most of the apps you are using on a day-to-day basis employ the HTTP protocol too; using a proxy can help you identify those weak spots

or even detect some hidden functionality or malicious behaviour. You can redirect traffic from your smartphone or IoT devices as easily as with any other computer and then feed it to Burp or ZAP! This way you will be able to intercept HTTP requests, manipulate them, and perform the usual sort of analysis and attacks as with any regular website.

Redirect (some) traffic

When it comes to redirecting, monitoring or even modifying HTTP requests from your devices, you will face two situations: either you have administrator privileges on the device or you don't. The former will allow you to configure the device network with your own under-your-control gateway or proxy (in case the device supports proxies, that is). The latter will leave you with two options:

#	Host	Meth...	URL	Para...	Edited	Status	Length	MIME ty...	Extensi...	Title
1	http://www.bicibox.cat	GET	/desktopmodules/biciboxsvc/b...	<input type="checkbox"/>	<input type="checkbox"/>	200	32404	JSON		
2	http://wservice.viabicing....	GET	/2/stations	<input type="checkbox"/>	<input type="checkbox"/>	200	1012...	JSON		
4	http://setting.rayjump.com	GET	/appwall/setting?app_id=3048...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	1212	JSON		
5	http://api.mobula.sdk.du...	GET	/adunion/slot/getAd?h=128...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	56326	JSON		
6	http://api.apptap.com	GET	/link/buy/android/com.privalla...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	302	660	HTML		
7	http://tracking.sumatoa...	GET	/aff_c?offer_id=167203&aff_id...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	302	2414	HTML		
8	http://tracking.sumatoa...	GET	/aff_r?offer_id=167203&aff_id...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	302	958	HTML		
9	http://apireporting.rvmm...	GET	/apiClick?cidRev=599b0fe142...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	303	317	text		
10	http://offernotavailable.r...	GET	/	<input type="checkbox"/>	<input type="checkbox"/>	200	1409	HTML		
12	http://api.videometrics.t...	GET	/ted_talks/trending?limit=100	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	4460	JSON		
14	http://check.googlezip.net	GET	/generate_204	<input type="checkbox"/>	<input type="checkbox"/>	204	102			
15	http://jcgghiaaorealy...	HEAD	/	<input type="checkbox"/>	<input type="checkbox"/>					
16	http://fhtruhuhhrw...	HEAD	/	<input type="checkbox"/>	<input type="checkbox"/>					

Request Response

Raw Headers Hex

```

X-Request-ID: 68cac7a5-af94-4904-813d-c7b4d15ca49e
X-Runtime: 0.519660
X-Varnish: 949484554 949482296
X-XSS-Protection: 1; mode=block
Content-Length: 3957
Connection: Close

{"meta":{"published_begin":null,"published_end":"2017-09-26T15:26:09Z","views_begin":"2017-10-24T15:26:09Z","views_end":null,"limit":100}, "ted_talks": [{"ted_id":848,"trending_views":31387}, {"ted_id":1569,"trending_views":30803}, {"ted_id":2861,"trending_views":27147}, {"ted_id":66,"trending_views":25148}, {"ted_id":1042,"trending_views":22065}, {"ted_id":2034,"trending_views":19916}, {"ted_id":2399,"trending_views":16144}, {"ted_id":1733,"trending_views":14964}, {"ted_id":2815,"trending_views":14806}, {"ted_id":2799,"trending_views":13318}, {"ted_id":2874,"trending_views":129893}, {"ted_id":2458,"trending_views":12979}, {"ted_id":2859,"trending_views":12919}, {"ted_id":1344,"trending_views":12314}, {"ted_id":652,"trending_views":12172}, {"ted_id":2867,"trending_views":11642}, {"ted_id":1654,"trending_views":11563}, {"ted_id":2299,"trending_views":10976}, {"ted_id":2884,"trending_views":10602}, {"ted_id":1815,"trending_views":10509}, {"ted_id":2872,"trending_views":10315}, {"ted_id":2835,"trending_views":10259}, {"ted_id":618,"trending_views":10218}, {"ted_id":2871,"trending_views":10089}, {"ted_id":2488,"trending_views":9795}, {"ted_id":1669,"trending_views":9509}, {"ted_id":2862,"trending_views":9378}, {"ted_id":2887,"trending_views":9333}, {"ted_id":2882,"trending_views":8956}, {"ted_id":2435,"trending_views":8759}, {"ted_id":2801,"trending_views":8342}, {"ted_id":2798,"trending_views":8321}, {"ted_id":1533,"trending_views":8225}, {"ted_id":2547,"trending_views":8122}, {"ted_id":1377,"trending_views":8008}, {"ted_id":2802,"trending_views":7929}, {"ted_id":1391,"trending_views":7639}, {"ted_id":70,"trending_views":7633}, {"ted_id":2870,"trending_views":7485}, {"ted_id":2879,"trending_views":7301}, {"ted_id":2869,"trending_views":7289}, {"ted_id":2852,"trending_views":7275}, {"ted_id":96,"trending_views":7192}, {"ted_id":2463,"trending_views":7159}, {"ted_id":2881,"trending_views":7144}, {"ted_id":1183,"trending_views":7144}, {"ted_id":2811,"trending_views":7144}, {"ted_id":2880,"trending_views":7144}, {"ted_id":2883,"trending_views":7144}, {"ted_id":2885,"trending_views":7144}, {"ted_id":2886,"trending_views":7144}, {"ted_id":2887,"trending_views":7144}, {"ted_id":2888,"trending_views":7144}, {"ted_id":2889,"trending_views":7144}, {"ted_id":2890,"trending_views":7144}, {"ted_id":2891,"trending_views":7144}, {"ted_id":2892,"trending_views":7144}, {"ted_id":2893,"trending_views":7144}, {"ted_id":2894,"trending_views":7144}, {"ted_id":2895,"trending_views":7144}, {"ted_id":2896,"trending_views":7144}, {"ted_id":2897,"trending_views":7144}, {"ted_id":2898,"trending_views":7144}, {"ted_id":2899,"trending_views":7144}, {"ted_id":2900,"trending_views":7144}, {"ted_id":2901,"trending_views":7144}, {"ted_id":2902,"trending_views":7144}, {"ted_id":2903,"trending_views":7144}, {"ted_id":2904,"trending_views":7144}, {"ted_id":2905,"trending_views":7144}, {"ted_id":2906,"trending_views":7144}, {"ted_id":2907,"trending_views":7144}, {"ted_id":2908,"trending_views":7144}, {"ted_id":2909,"trending_views":7144}, {"ted_id":2910,"trending_views":7144}, {"ted_id":2911,"trending_views":7144}, {"ted_id":2912,"trending_views":7144}, {"ted_id":2913,"trending_views":7144}, {"ted_id":2914,"trending_views":7144}, {"ted_id":2915,"trending_views":7144}, {"ted_id":2916,"trending_views":7144}, {"ted_id":2917,"trending_views":7144}, {"ted_id":2918,"trending_views":7144}, {"ted_id":2919,"trending_views":7144}, {"ted_id":2920,"trending_views":7144}, {"ted_id":2921,"trending_views":7144}, {"ted_id":2922,"trending_views":7144}, {"ted_id":2923,"trending_views":7144}, {"ted_id":2924,"trending_views":7144}, {"ted_id":2925,"trending_views":7144}, {"ted_id":2926,"trending_views":7144}, {"ted_id":2927,"trending_views":7144}, {"ted_id":2928,"trending_views":7144}, {"ted_id":2929,"trending_views":7144}, {"ted_id":2930,"trending_views":7144}, {"ted_id":2931,"trending_views":7144}, {"ted_id":2932,"trending_views":7144}, {"ted_id":2933,"trending_views":7144}, {"ted_id":2934,"trending_views":7144}, {"ted_id":2935,"trending_views":7144}, {"ted_id":2936,"trending_views":7144}, {"ted_id":2937,"trending_views":7144}, {"ted_id":2938,"trending_views":7144}, {"ted_id":2939,"trending_views":7144}, {"ted_id":2940,"trending_views":7144}, {"ted_id":2941,"trending_views":7144}, {"ted_id":2942,"trending_views":7144}, {"ted_id":2943,"trending_views":7144}, {"ted_id":2944,"trending_views":7144}, {"ted_id":2945,"trending_views":7144}, {"ted_id":2946,"trending_views":7144}, {"ted_id":2947,"trending_views":7144}, {"ted_id":2948,"trending_views":7144}, {"ted_id":2949,"trending_views":7144}, {"ted_id":2950,"trending_views":7144}, {"ted_id":2951,"trending_views":7144}, {"ted_id":2952,"trending_views":7144}, {"ted_id":2953,"trending_views":7144}, {"ted_id":2954,"trending_views":7144}, {"ted_id":2955,"trending_views":7144}, {"ted_id":2956,"trending_views":7144}, {"ted_id":2957,"trending_views":7144}, {"ted_id":2958,"trending_views":7144}, {"ted_id":2959,"trending_views":7144}, {"ted_id":2960,"trending_views":7144}, {"ted_id":2961,"trending_views":7144}, {"ted_id":2962,"trending_views":7144}, {"ted_id":2963,"trending_views":7144}, {"ted_id":2964,"trending_views":7144}, {"ted_id":2965,"trending_views":7144}, {"ted_id":2966,"trending_views":7144}, {"ted_id":2967,"trending_views":7144}, {"ted_id":2968,"trending_views":7144}, {"ted_id":2969,"trending_views":7144}, {"ted_id":2970,"trending_views":7144}, {"ted_id":2971,"trending_views":7144}, {"ted_id":2972,"trending_views":7144}, {"ted_id":2973,"trending_views":7144}, {"ted_id":2974,"trending_views":7144}, {"ted_id":2975,"trending_views":7144}, {"ted_id":2976,"trending_views":7144}, {"ted_id":2977,"trending_views":7144}, {"ted_id":2978,"trending_views":7144}, {"ted_id":2979,"trending_views":7144}, {"ted_id":2980,"trending_views":7144}, {"ted_id":2981,"trending_views":7144}, {"ted_id":2982,"trending_views":7144}, {"ted_id":2983,"trending_views":7144}, {"ted_id":2984,"trending_views":7144}, {"ted_id":2985,"trending_views":7144}, {"ted_id":2986,"trending_views":7144}, {"ted_id":2987,"trending_views":7144}, {"ted_id":2988,"trending_views":7144}, {"ted_id":2989,"trending_views":7144}, {"ted_id":2990,"trending_views":7144}, {"ted_id":2991,"trending_views":7144}, {"ted_id":2992,"trending_views":7144}, {"ted_id":2993,"trending_views":7144}, {"ted_id":2994,"trending_views":7144}, {"ted_id":2995,"trending_views":7144}, {"ted_id":2996,"trending_views":7144}, {"ted_id":2997,"trending_views":7144}, {"ted_id":2998,"trending_views":7144}, {"ted_id":2999,"trending_views":7144}, {"ted_id":3000,"trending_views":7144}, {"ted_id":3001,"trending_views":7144}, {"ted_id":3002,"trending_views":7144}, {"ted_id":3003,"trending_views":7144}, {"ted_id":3004,"trending_views":7144}, {"ted_id":3005,"trending_views":7144}, {"ted_id":3006,"trending_views":7144}, {"ted_id":3007,"trending_views":7144}, {"ted_id":3008,"trending_views":7144}, {"ted_id":3009,"trending_views":7144}, {"ted_id":3010,"trending_views":7144}, {"ted_id":3011,"trending_views":7144}, {"ted_id":3012,"trending_views":7144}, {"ted_id":3013,"trending_views":7144}, {"ted_id":3014,"trending_views":7144}, {"ted_id":3015,"trending_views":7144}, {"ted_id":3016,"trending_views":7144}, {"ted_id":3017,"trending_views":7144}, {"ted_id":3018,"trending_views":7144}, {"ted_id":3019,"trending_views":7144}, {"ted_id":3020,"trending_views":7144}, {"ted_id":3021,"trending_views":7144}, {"ted_id":3022,"trending_views":7144}, {"ted_id":3023,"trending_views":7144}, {"ted_id":3024,"trending_views":7144}, {"ted_id":3025,"trending_views":7144}, {"ted_id":3026,"trending_views":7144}, {"ted_id":3027,"trending_views":7144}, {"ted_id":3028,"trending_views":7144}, {"ted_id":3029,"trending_views":7144}, {"ted_id":3030,"trending_views":7144}, {"ted_id":3031,"trending_views":7144}, {"ted_id":3032,"trending_views":7144}, {"ted_id":3033,"trending_views":7144}, {"ted_id":3034,"trending_views":7144}, {"ted_id":3035,"trending_views":7144}, {"ted_id":3036,"trending_views":7144}, {"ted_id":3037,"trending_views":7144}, {"ted_id":3038,"trending_views":7144}, {"ted_id":3039,"trending_views":7144}, {"ted_id":3040,"trending_views":7144}, {"ted_id":3041,"trending_views":7144}, {"ted_id":3042,"trending_views":7144}, {"ted_id":3043,"trending_views":7144}, {"ted_id":3044,"trending_views":7144}, {"ted_id":3045,"trending_views":7144}, {"ted_id":3046,"trending_views":7144}, {"ted_id":3047,"trending_views":7144}, {"ted_id":3048,"trending_views":7144}, {"ted_id":3049,"trending_views":7144}, {"ted_id":3050,"trending_views":7144}, {"ted_id":3051,"trending_views":7144}, {"ted_id":3052,"trending_views":7144}, {"ted_id":3053,"trending_views":7144}, {"ted_id":3054,"trending_views":7144}, {"ted_id":3055,"trending_views":7144}, {"ted_id":3056,"trending_views":7144}, {"ted_id":3057,"trending_views":7144}, {"ted_id":3058,"trending_views":7144}, {"ted_id":3059,"trending_views":7144}, {"ted_id":3060,"trending_views":7144}, {"ted_id":3061,"trending_views":7144}, {"ted_id":3062,"trending_views":7144}, {"ted_id":3063,"trending_views":7144}, {"ted_id":3064,"trending_views":7144}, {"ted_id":3065,"trending_views":7144}, {"ted_id":3066,"trending_views":7144}, {"ted_id":3067,"trending_views":7144}, {"ted_id":3068,"trending_views":7144}, {"ted_id":3069,"trending_views":7144}, {"ted_id":3070,"trending_views":7144}, {"ted_id":3071,"trending_views":7144}, {"ted_id":3072,"trending_views":7144}, {"ted_id":3073,"trending_views":7144}, {"ted_id":3074,"trending_views":7144}, {"ted_id":3075,"trending_views":7144}, {"ted_id":3076,"trending_views":7144}, {"ted_id":3077,"trending_views":7144}, {"ted_id":3078,"trending_views":7144}, {"ted_id":3079,"trending_views":7144}, {"ted_id":3080,"trending_views":7144}, {"ted_id":3081,"trending_views":7144}, {"ted_id":3082,"trending_views":7144}, {"ted_id":3083,"trending_views":7144}, {"ted_id":3084,"trending_views":7144}, {"ted_id":3085,"trending_views":7144}, {"ted_id":3086,"trending_views":7144}, {"ted_id":3087,"trending_views":7144}, {"ted_id":3088,"trending_views":7144}, {"ted_id":3089,"trending_views":7144}, {"ted_id":3090,"trending_views":7144}, {"ted_id":3091,"trending_views":7144}, {"ted_id":3092,"trending_views":7144}, {"ted_id":3093,"trending_views":7144}, {"ted_id":3094,"trending_views":7144}, {"ted_id":3095,"trending_views":7144}, {"ted_id":3096,"trending_views":7144}, {"ted_id":3097,"trending_views":7144}, {"ted_id":3098,"trending_views":7144}, {"ted_id":3099,"trending_views":7144}, {"ted_id":3100,"trending_views":7144}, {"ted_id":3101,"trending_views":7144}, {"ted_id":3102,"trending_views":7144}, {"ted_id":3103,"trending_views":7144}, {"ted_id":3104,"trending_views":7144}, {"ted_id":3105,"trending_views":7144}, {"ted_id":3106,"trending_views":7144}, {"ted_id":3107,"trending_views":7144}, {"ted_id":3108,"trending_views":7144}, {"ted_id":3109,"trending_views":7144}, {"ted_id":3110,"trending_views":7144}, {"ted_id":3111,"trending_views":7144}, {"ted_id":3112,"trending_views":7144}, {"ted_id":3113,"trending_views":7144}, {"ted_id":3114,"trending_views":7144}, {"ted_id":3115,"trending_views":7144}, {"ted_id":3116,"trending_views":7144}, {"ted_id":3117,"trending_views":7144}, {"ted_id":3118,"trending_views":7144}, {"ted_id":3119,"trending_views":7144}, {"ted_id":3120,"trending_views":7144}, {"ted_id":3121,"trending_views":7144}, {"ted_id":3122,"trending_views":7144}, {"ted_id":3123,"trending_views":7144}, {"ted_id":3124,"trending_views":7144}, {"ted_id":3125,"trending_views":7144}, {"ted_id":3126,"trending_views":7144}, {"ted_id":3127,"trending_views":7144}, {"ted_id":3128,"trending_views":7144}, {"ted_id":3129,"trending_views":7144}, {"ted_id":3130,"trending_views":7144}, {"ted_id":3131,"trending_views":7144}, {"ted_id":3132,"trending_views":7144}, {"ted_id":3133,"trending_views":7144}, {"ted_id":3134,"trending_views":7144}, {"ted_id":3135,"trending_views":7144}, {"ted_id":3136,"trending_views":7144}, {"ted_id":3137,"trending_views":7144}, {"ted_id":3138,"trending_views":7144}, {"ted_id":3139,"trending_views":7144}, {"ted_id":3140,"trending_views":7144}, {"ted_id":3141,"trending_views":7144}, {"ted_id":3142,"trending_views":7144}, {"ted_id":3143,"trending_views":7144}, {"ted_id":3144,"trending_views":7144}, {"ted_id":3145,"trending_views":7144}, {"ted_id":3146,"trending_views":7144}, {"ted_id":3147,"trending_views":7144}, {"ted_id":3148,"trending_views":7144}, {"ted_id":3149,"trending_views":7144}, {"ted_id":3150,"trending_views":7144}, {"ted_id":3151,"trending_views":7144}, {"ted_id":3152,"trending_views":7144}, {"ted_id":3153,"trending_views":7144}, {"ted_id":3154,"trending_views":7144}, {"ted_id":3155,"trending_views":7144}, {"ted_id":3156,"trending_views":7144}, {"ted_id":3157,"trending_views":7144}, {"ted_id":3158,"trending_views":7144}, {"ted_id":3159,"trending_views":7144}, {"ted_id":3160,"trending_views":7144}, {"ted_id":3161,"trending_views":7144}, {"ted_id":3162,"trending_views":7144}, {"ted_id":3163,"trending_views":7144}, {"ted_id":3164,"trending_views":7144}, {"ted_id":3165,"trending_views":7144}, {"ted_id":3166,"trending_views":7144}, {"ted_id":3167,"trending_views":7144}, {"ted_id":3168,"trending_views":7144}, {"ted_id":3169,"trending_views":7144}, {"ted_id":3170,"trending_views":7144}, {"ted_id":3171,"trending_views":7144}, {"ted_id":3172,"trending_views":7144}, {"ted_id":3173,"trending_views":7144}, {"ted_id":3174,"trending_views":7144}, {"ted_id":3175,"trending_views":7144}, {"ted_id":3176,"trending_views":7144}, {"ted_id":3177,"trending_views":7144}, {"ted_id":3178,"trending_views":7144}, {"ted_id":3179,"trending_views":7144}, {"ted_id":3180,"trending_views":7144}, {"ted_id":3181,"trending_views":7144}, {"ted_id":3182,"trending_views":7144}, {"ted_id":3183,"trending_views":7144}, {"ted_id":3184,"trending_views":7144}, {"ted_id":3185,"trending_views":7144}, {"ted_id":3186,"trending_views":7144}, {"ted_id":3187,"trending_views":7144}, {"ted_id":3188,"trending_views":7144}, {"ted_id":3189,"trending_views":7144}, {"ted_id":3190,"trending_views":7144}, {"ted_id":3191,"trending_views":7144}, {"ted_id":3192,"trending_views":7144}, {"ted_id":3193,"trending_views":7144}, {"ted_id":3194,"trending_views":7144}, {"ted_id":3195,"trending_views":7144}, {"ted_id":3196,"trending_views":7144}, {"ted_id":3197,"trending_views":7144}, {"ted_id":3198,"trending_views":7144}, {"ted_id":3199,"trending_views":7144}, {"ted_id":3200,"trending_views":7144}, {"ted_id":3201,"trending_views":7144}, {"ted_id":3202,"trending_views":7144}, {"ted_id":3203,"trending_views":7144}, {"ted_id":3204,"trending_views":7144}, {"ted_id":3205,"trending_views":7144}, {"ted_id":3206,"trending_views":7144}, {"ted_id":3207,"trending_views":7144}, {"ted_id":3208,"trending_views":7144}, {"ted_id":3209,"trending_views":7144}, {"ted_id":3210,"trending_views":7144}, {"ted_id":3211,"trending_views":7144}, {"ted_id":3212,"trending_views":7144}, {"ted_id":3213,"trending_views":7144}, {"ted_id":3214,"trending_views":7144}, {"ted_id":3215,"trending_views":7144}, {"ted_id":3216,"trending_views":7144}, {"ted_id":3217,"trending_views":7144}, {"ted_id":3218,"trending_views":7144}, {"ted_id":3219,"trending_views":7144}, {"ted_id":3220,"trending_views":7144}, {"ted_id":3221,"trending_views":7144}, {"ted_id":3222,"trending_views":7144}, {"ted_id":3223,"trending_views":7144}, {"ted_id":3224,"trending_views":7144}, {"ted_id":3225,"trending_views":7144}, {"ted_id":3226,"trending_views":7144}, {"ted_id":3227,"trending_views":7144}, {"ted_id":3228,"trending_views":7144}, {"ted_id":3229,"trending_views":7144}, {"ted_id":3230,"trending_views":7144}, {"ted_id":3231,"trending_views":7144}, {"ted_id":3232,"trending_views":7144}, {"ted_id":3233,"trending_views":7144}, {"ted_id":3234,"trending_views":7144}, {"ted_id":3235,"trending_views":7144}, {"ted_id":3236,"trending_views":7144}, {"ted_id":3237,"trending_views":7144}, {"ted_id":3238,"trending_views":7144}, {"ted_id":3239,"trending_views":7144}, {"ted_id":3240,"trending_views":7144}, {"ted_id":3241,"trending_views":7144}, {"ted_id":3242,"trending_views":7144}, {"ted_id":3243,"trending_views":7144}, {"ted_id":3244,"trending_views":7144}, {"ted_id":3245,"trending_views":7144}, {"ted_id":3246,"trending_views":7144}, {"ted_id":3247,"trending_views":7144}, {"ted_id":3248,"trending_views":7144}, {"ted_id":3249,"trending_views":7144}, {"ted_id":3250,"trending_views":7144}, {"ted_id":3251,"trending_views":7144}, {"ted_id":3252,"trending_views":7144}, {"ted_id":3253,"trending_views":7144}, {"ted_id":3254,"trending_views":7144}, {"ted_id":3255,"trending_views":7144}, {"ted_id":3256,"trending_views":7144}, {"ted_id":3257,"trending_views":7144}, {"ted_id":3258,"trending_views":7144}, {"ted_id":3259,"trending_views":7144}, {"ted_id":3260,"trending_views":7144}, {"ted_id":3261,"trending_views":7144}, {"ted_id":3262,"trending_views":7144}, {"ted_id":3263,"trending_views":7144}, {"ted_id":3264,"trending_views":7144}, {"ted_id":3265,"trending_views":7144}, {"ted_id":3266,"trending_views":7144}, {"ted_id":3267,"trending_views":7144}, {"ted_id":3268,"trending_views":7144}, {"ted_id":3269,"trending_views":7144}, {"ted_id":3270,"trending_views":7144}, {"ted_id":3271,"trending_views":7144}, {"ted_id":3272,"trending_views":7144}, {"ted_id":3273,"trending_views":7144}, {"ted_id":3274,"trending_views":7144}, {"ted_id":3275,"trending_views":7144}, {"ted_id":3276,"trending_views":7144}, {"ted_id":3277,"trending_views":7144}, {"ted_id":3278,"trending_views":7144}, {"ted_id":3279,"trending_views":7144}, {"ted_id":3280,"trending_views":7144}, {"ted_id":3281,"trending_views":7144}, {"ted_id":3282,"trending_views":7144}, {"ted_id":3283,"trending_views":7144}, {"ted_id":3284,"trending_views":7144}, {"ted_id":3285,"trending_views":7144}, {"ted_id":3286,"trending_views":7144}, {"ted_id":3287,"trending_views":7144}, {"ted_id":3288,"trending_views":7144}, {"ted_id":3289,"trending_views":7144}, {"ted_id":3290,"trending_views":7144}, {"ted_id":3291,"trending_views":7144}, {"ted_id":3292,"trending_views":7144}, {"ted_id":3293,"trending_views":7144}, {"ted_id":3294,"trending_views":7144}, {"ted_id":329
```

either you can redirect traffic by setting up some rules on your network router (assuming you do have administrator rights on the network router, of course) or you poison the device's ARP cache with an 'ARP spoofing attack'. Let's focus first on devices for which you have admin access. Most IoT devices that use Wi-Fi functionality to communicate with the network can be automatically configured by means of the DHCP service. Some others may allow you to set up their TCP/IP settings manually. Android devices can be configured to use a proxy server as well: follow the next steps in order to intercept any HTTP request from your smartphone using Burp (bear in mind that these steps may vary depending on your Android version and device vendor):

1. On your GNU/Linux computer, execute Burp and go to the Proxy/Options tab.
2. Push the 'Add' button below 'Proxy Listeners'. Type 8080 in the 'Bind to port' text box and then select your private IP in the drop-down list 'Specific address' (it should be something like 192.168.X.X). Finally, press the 'OK' button. A new proxy listener will be shown.
3. Make sure to disable traffic interception by pushing the 'Intercept is ON' button on the Intercept tab. Then go to the HTTP History tab.
4. On your Android device, go to Settings > Wi-Fi. Long-press your Wi-Fi ESSID and select 'Modify network'.
5. Enable the checkbox 'Advanced options' and scroll down a bit until you find the Proxy drop-down list widget.
6. Tap it and select 'Manual'.
7. Type your computer's IP address in the 'Proxy hostname' text box, 8080 in the 'Proxy port' text box and then tap 'Save'.

Now open your favourite web browser on your smartphone (e.g. Google Chrome) and browse to any non-SSL website. You will see some activity on the 'Proxy/HTTP history' tab in Burp. As clearly stated by Android: "The HTTP proxy(... may not be used by the other apps".

“ You can redirect traffic from your smartphone or IoT devices as easily as with any other computer ”

Try it: toy with some apps that make use of the internet, such as Ted Talks and the Merriam-Webster dictionary; for some you will see HTTP requests going through Burp whereas for some others you won't. If you try to connect to TLS-protected websites, however, you will get an error message because of a certificate mismatch. You can fix that by installing a Burp SSL certificate on your device:

1. In Burp, go to the Proxy/Options tab and click the 'Import/Export CA certificate' button.
2. Select 'Export/Certificate in DER format' and then click 'Next'.

3. Save the file somewhere in your computer; name it as **burp.crt**.

4. Now, upload this file to your Android device (e.g. by means of the **adb** command).

5. On your Android device, tap Settings > Security > 'Install from (phone) storage'. Choose **burp.crt** as the new certificate to install. Give it a name (e.g. BURP) and then tap 'OK'.

6. If you tap Settings > Security > Trusted credentials/user, you will see the new installed certificate. Tap it to get some information (such as its fingerprint and validity).

If you navigate to TLS-protected websites this time, no security warnings will be shown at all. Besides, the HTTP History tab in Burp will show you all the TLS traffic in cleartext (i.e. decrypted). This is so because Burp is now performing a proper 'man-in-the-middle' (MITM) attack. Make sure to remove this certificate once you are done with this tutorial. Attackers who might be able to get their hands on its private key would decrypt your TLS connections just as easily.



Above How miserable: you've just managed to perform a MITM attack against yourself!

Redirect (all) traffic

So far so good but... what about those apps that do not use the proxy? Not to mention those IoT devices that do not have an option to set up a proxy in the first place! Enter iptables and IP forwarding! Open a new terminal on your GNU/Linux computer and enable IP forwarding first:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Get back to your Android device and note its current IP address: Settings > Wi-Fi > Advanced. You are going to set up your computer as a router for your smartphone by using the iptables FORWARD chain. In order to forward any packet from/to your device, type the following commands (replace <ethX> with the network device that is connected to the same network as your Android device and <IP_ANDROID> with your phone's IP address):

```
# iptables -t nat -I POSTROUTING 1 -o <ethX> -j MASQUERADE
# iptables -I FORWARD 1 -o <ethX> -s <IP_ANDROID> -j ACCEPT
# iptables -I FORWARD 2 -i <ethX> -d <IP_ANDROID> -j ACCEPT
```

Please note that if you have the default policy of the

Smart firewalls

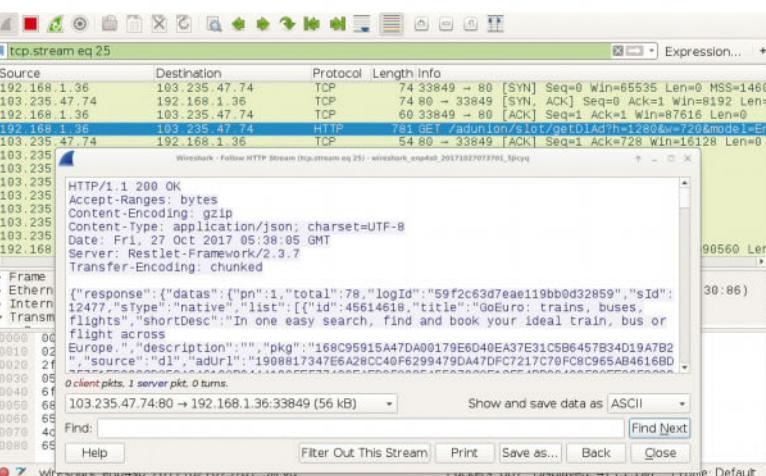
With all the threats IoT devices have to face, some vendors have started to develop a new generation of smart firewalls. These devices monitor the network and – thanks to their cloud infrastructure, big data and deep learning techniques – they can spot suspicious activity, send warnings or even take a proactive approach by blocking packets. Check out these examples: Cujo, Norton Core, RATtrap and Bitdefender Box.

■ Burp and SSLStrip-like attacks

For those devices that are using TLS to protect their communications, you will not be able to decrypt their packets by just ARP-poisoning their caches. SSLstrip to the rescue! Surprisingly enough, and as long as the HSTS header is not present and the first connection is through HTTP, this attack still delivers. Burp includes some options in order to perform SSLstrip-like attacks, too (see <http://bit.ly/SSLstrip>).



Below Maybe you don't see them right away, but ads are always a few HTTP GETs away!



FORWARD chain set to 'ACCEPT', you don't even need to add the last two rules at all. We encourage you to set your default firewall policy to DROP always, though. Now your computer is ready to start forwarding packets from/to your device. Next, set up the gateway manually on your smartphone:

1. Go to Settings > Wi-Fi, long-press your Wi-Fi ESSID and select 'Modify network'.
2. Enable 'Advanced options' and tap 'IPv4 settings'. Select 'Static', then type the original IP address of this device in the 'IPv4 address' text box and your computer's IP address in the Gateway text-box.
3. Tap 'Proxy' and select 'None' to disable the proxy. Finally, tap 'Save'.

You can open any app and use it the normal way: this time, however, instead of using your router/AP in order to access the internet, the entire system will be routing all the network traffic through your computer. You can start capturing some traffic right away with Wireshark: execute it and select the network adaptor that has the IP address you have set up on your Android device as its gateway. In order to capture only packets coming from or going to your smartphone, set this capture filter: 'host <IP_ANDROID>' and disable the 'promiscuous mode'. Start using some apps (such as Telegram and your web browser) and you'll see a lot of packets being captured by Wireshark. All TLS packets will be encrypted, though. This is because you are not forwarding these packets to Burp proxy. Therefore, they cannot be decrypted because your computer is merely forwarding packets from/to your device dumbly and capturing them along the way. So the next step is to forward some of these packets transparently to Burp. Let's imagine you want to see every packet destined to or received from ports 443/TCP and 80/TCP. On your GNU/Linux computer, type this in the terminal:

```
# iptables -t nat -A PREROUTING -p tcp
--dport 443 -j REDIRECT --to-port 8080
# iptables -t nat -A PREROUTING -p tcp
```

```
--dport 80 -j REDIRECT --to-port 8080
```

Execute Burp and make sure it is listening on port 8080/TCP. Because now you want to redirect every HTTP request to ports 443 and 80 transparently, you have to set up Burp proxy in 'transparent' mode. Otherwise apps won't be able to connect to the internet because they are not aware of the proxy at all (this sort of proxy is known as a 'Transparent Proxy'):

1. In Burp, go to the Proxy/Options tab.
2. Select the proxy listener you have configured previously and click 'Edit'.
3. Go to the Request Handling tab and enable the option 'Support invisible proxying(...)' Finally, hit the 'Save' button.

Get back to your Android device and make sure the Burp certificate is installed. Then, execute some apps that make use of TLS encryption (e.g. LinkedIn). The HTTP History tab in Burp will show you every single HTTP

“ The HTTP History tab in Burp will show you all the TLS traffic in cleartext, i.e. decrypted ”

request and response in cleartext. Some apps may check the server certificate; for these apps the connection will irremediably fail. Of course, there may be apps that use different ports for their HTTP requests; by using the PREROUTING chain in the NAT table, you will be able to feed Burp anyway. If you do not know beforehand which port a particular app is using, open Wireshark and capture some traffic. After a while, you will have some packets from your device sent to an external IP address on a certain port. Then you can add a new rule to the PREROUTING chain for that particular port.

Forwarding packets allows you to detect unusual patterns on your devices, too (what follows is the cheap way; for some advanced protection, see Smart Firewalls box, p47). Imagine that you are monitoring all the packets coming from your new gadget (a smart light-bulb). When the device is working normally, you see some HTTP activity encapsulating some JSON data coming back and forth between your smart light-bulb's IP address and a remote server in the cloud (e.g. Amazon S3). This is its normal behaviour, so to speak. So you set a capture filter such as: `src host <LIGHTBULB_IP>` and leave Wireshark capturing data for a while. Later on, you use the Statistics > IPv4 Statistics > 'Source and Destination Addresses' menu option and you find out that, besides the remote Amazon S3 server you consider legit, there is another IP address you haven't seen before. Its count column (number of packets sent to this destination address) is worryingly high. You navigate to Robtex

(www.robtex.com) and paste this IP address in the search text box. Robtex tells you that this an IP from Russia. You do your research and, apparently, this remote IP has nothing to do neither with your vendor's light-bulb nor with the cloud service it uses. The domain it belongs to is reportedly malicious. Indeed, your smart gadget has been p0wned! Congrats: now it is a new zombie working for the zillionth botnet!

Perform MITM attacks

How about those devices you cannot control? Let's imagine the device is connected to a network you don't manage; setting up some rules on the network router is out of the question, too. Next thing could be to set up your own rogue DHCP server, but maybe the device's TCP/IP settings have been configured as static. So what's next? Performing a classic MITM attack by spoofing its ARP cache surely won't hurt. This way all its traffic will be redirected to your computer and you will become its gateway instead of the actual one on the network. Sometimes this won't work at all or it may work partially. On some occasions, the gateway may have a static ARP entry for particular devices (the other way round is frankly unlikely). So let's imagine you want to impersonate the gateway (192.168.1.1) in order to capture all the traffic coming from your device 'DEVICE'. Execute the `arpspoof` command this way:

```
# arpspoof -i <ethX> -t <IP_DEVICE> 192.168.1.1 2> /dev/null&
# arpspoof -i <ethX> -t 192.168.1.1 <IP_DEVICE> 2>/dev/null&
```

Now you can run Wireshark and you will see all the packets from DEVICE. Of course you can redirect some of these packets to Burp as well. Do it now: make sure to add the desired rules to redirect any non-SSL HTTP

“Forwarding packets allows you to detect unusual patterns on your devices, too”

request to Burp as seen before. If you want to redirect the traffic to some other computer under your control, that is possible too. Use DNAT instead of REDIRECT:

```
# iptables -t nat -A PREROUTING
-p tcp --dport 80 -j DNAT --to-
destination <ANOTHER_IP>:8080
```

And don't forget to start Burp on that computer too, listening on port 8080/TCP. Once you are done with the poisoning, kill both arpspoof processes by executing `killall arpspoof`.

Intercept non-HTTP traffic

Burp deals with HTTP and HTTPS requests. But of course, some devices may be using their own sort of protocol to communicate. Josh Summit has developed a Burp extension that allows non-HTTP interception within Burp: NoPE Proxy (<http://bit.ly/NopeProxy>). Install telnetd on your GNU/Linux now: `apt-get install telnetd`. Then, download and install NoPE Proxy (see Resources, p46). Open Burp and go to the Extender tab. Push the 'Add' button and then 'Select File' besides 'Extension File (.jar)' and select `NopeProxy.jar`. Then click 'Next'. Once the extension is successfully installed, go to the NoPE Proxy/Server Settings tab. You are about to intercept Telnet packets so, under 'Not HTTP Proxy settings', type your computer's IP address in 'Server Address', 23 in 'Server Port', 9999 in 'Listening Port' and leave 'Certificate Server' empty. Finally, push the big green '+' button to add this new listener to the list below. Make sure to enable this listener by clicking on the 'Enable' checkbox. Now open a new terminal and redirect port 23/TCP to this listener:

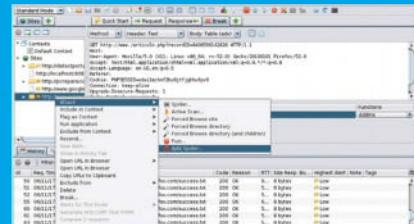
```
# iptables -t nat -A PREROUTING -p
tcp --dport 23 -j REDIRECT --to-port
9999
```

Next, install a Telnet client on your smartphone and make sure that the MITM attack is still going on against your device. Then, try to connect to your computer using Telnet. You will see some packets on the TCP History tab in Burp. Once you are presented with the 'login' prompt, and before typing

any username, push the 'Intercept is OFF' button on the TCP Intercept tab. On your device, type any login and send it to the server. You will see a new Telnet packet on the TCP Intercept tab waiting to be manipulated before being sent for real. Change the login string to something else by editing the text right in this tab and then disable 'Intercept is ON'. Finally, forward the packet by pushing the '>>' button. You will see how the server is now expecting a password for this new login! ■

WHAT NEXT?

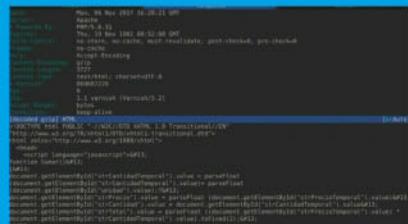
Alternatives to Burp



1 OWASP Zap! Proxy

The Zed Attack Proxy (ZAP) from OWASP is an incredible and powerful open source proxy quite similar in functionality to Burp. It ships with a lot of automated tools to scan for vulnerabilities in web applications and it can be automated thanks to its powerful REST API.

<http://bit.ly/OWASP-ZAP>



2 mitmproxy

This is an open source MITM proxy that allows HTTP and HTTPS interception with an ncurses-like graphical interface. If you are a fan of tcpdump, try mitmdump instead. You can install it right away on your Debian-based distro: `apt-get install mitmproxy`.

<https://mitmproxy.org>

3 TamperData

An outdated Firefox add-on, but still functional. It is as simple as it gets: it can intercept and manipulate POST parameters. It is quite unstable, but perfect for quick-and-dirty tests on-the-fly. <http://bit.ly/TamperData>

4 Charles Proxy

Another MITM HTTP and HTTPS proxy that includes some interesting functionalities such as bandwidth throttling to simulate slower internet connections, a full AJAX-debugger engine and support for AMF 0 and AMF 3 parsing.

www.charlesproxy.com/download

5 Telerik Fiddler

A MITM proxy too (it can decrypt HTTPS requests), it has been designed to work on Windows OSes although thanks to the Mono framework it can also work on GNU/Linux. It also supports extensions developed using any .NET language.

www.telerik.com/fiddler



Alexander Smith

is a computational physicist. He teaches Arduino to grad students and discourages people from doing lab work manually.

Resources

■ Arduino /
Raspberry Pi / Computer to receive

■ Arduino Nano (or barebones) to monitor

■ Temperature Sensor: DHT22

■ Radio transmitter & receiver (or Bluetooth, Wi-Fi adaptor for Arduino)

■ Battery power (a power bank, coin batteries or AA batteries and case)

■ Adafruit Sensor library
<http://bit.ly/AdafruitSensor>

■ Virtual Wire library
<http://bit.ly/VirtualWire>

■ Python
www.python.org

Use Arduino's sleep mode to monitor temperature

Protect your pipes and heat your home effectively by monitoring the temperature over the season

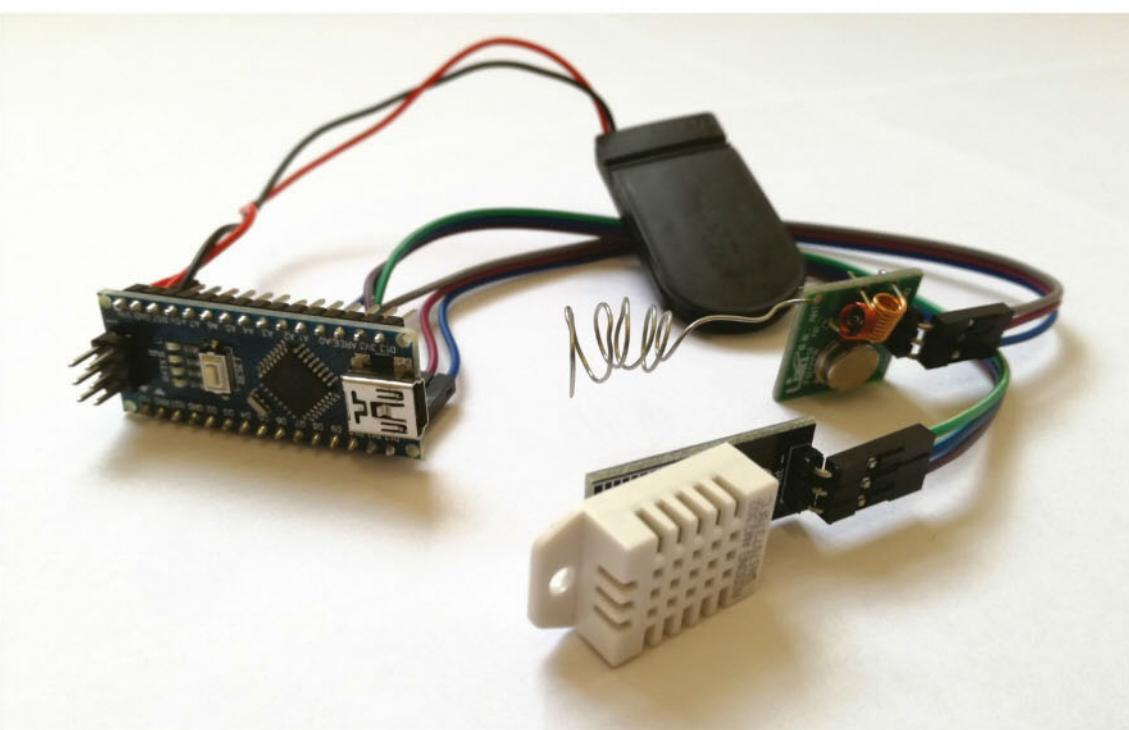
Temperatures are beginning to drop to freezing and the days are already short. For those worried about spending all their money on running the boiler, January is often the most agitating month. You've already bled your radiators and are resigned to wearing jumpers, but are still concerned about the pipes freezing and bursting. In this issue of **LU&D**, we'll be creating an Arduino temperature monitor which can be placed anywhere in the house and run for months on end, automatically reporting its data back to base so that you can make informed decisions about when to turn the heating on.

We'll begin by taking a lower-power Arduino and preparing the temperature sensor. In **LU&D183** we used an LM35, so in this tutorial we'll show you how to interface with the DHT22 using a library provided by Adafruit. We'll then add the radio transmitter to send messages containing the measurements, and create a base station to receive the data. Finally, we'll utilise the Atmel ATmega's sleep mode, waking up every so often

and powering the board only when required so that the battery doesn't drain all at once. In the end, we'll produce a temperature monitor which can run for the whole season without any maintenance, and gather incredibly rich data about temperature cycle of the coldest room in your home. From this you should be able to tell if you really should be turning your radiators turn on (or hopefully off) a few hours earlier.

For this tutorial, your choice of Arduino matters substantially. A battery only has so much capacity, measured in milliamp hours (mAh). Most small power banks deliver 5V with a capacity of around 2200mAh. An Arduino Uno will use about 45mAh in idle mode, which equates to about 48 hours of operation before it powers down. This consumption is due to the large number of hardware components sitting on the board, all of which require power even if they are not being actively used.

To reduce this burden on a battery, it's a good idea to make your own Arduino from parts and with as few



Above Arduino Nano, with temperature sensor, transmitter and coin battery

components as is required. These ‘bare-bones’ boards can eliminate two-thirds of the power wastage and open the possibility of operating the microcontroller in microamps of current instead – potentially allowing operation for years. If this is beyond your skill set, we recommend using a smaller board, such as the Arduino Pro which ships with less hardware. However, for this demonstration, we’re going to be using the Arduino Nano and it should be able to run for a few weeks before needing a change of battery.

Prepare the sensor

The DHT22 is a temperature and humidity sensor, available cheaply online. It comes in two variants, one with a circuit board attached and one without. If you purchase the version without the pre-built circuit, you will need to add a pull-up resistor between the voltage supply and the data pins, as communication is performed by creating a drop in voltage between data and ground. The data sheet suggests a 1k resistor, but it’s not uncommon to see a 10k resistor online. Connect voltage-in, ground and data to the digital pins on your Arduino, but avoid the pulse-width modulated pins – denoted by a tilde (~).

To interface with the sensor, we are going to use the Adafruit Sensor library and DHT Sensor library. Both are available for download through the Arduino IDE, which also handles the installation. This saves us from having to write our own program to deal with the communication protocol between the microcontroller and the DHT22. Briefly, the Arduino sends a drop in voltage to the sensor to say it’s waiting for data. The sensor then confirms it received the instruction and then sends a stream of 8 bits for each reading. The Adafruit library handles all this for us and deals with the pulse timings.

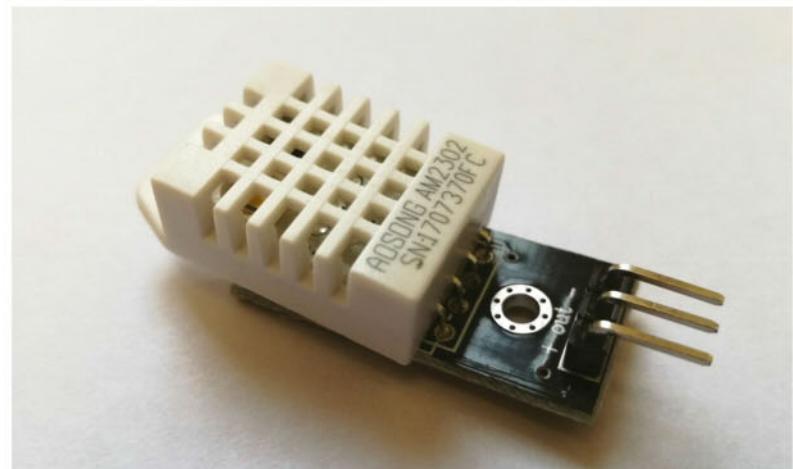
It’s time to start programming! Open the Arduino IDE and create a new sketch. To set up the temperature sensor, we need to include a few lines at the top of the code so that the library knows how to handle the sensor and which pin to use for communication. Make sure DHTPIN is set to the data pin connected to your Arduino. We then create a DHT object which provides methods that handle the communication and conversion of data:

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```

In ‘setup’, we then need to call `dht.begin()` to initialise the DHT object and then set the digital pins to 5V. Then only one method is needed to get a reading for temperature and another for humidity which can be performed in the loop function:

```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```

The sensor will take a couple of seconds to acquire a measurement and also needs two seconds to ‘warm



Above DHT22 – sensor for measuring temperature and humidity

up’ after supplying power before it can be used. This point will become important later when we begin to turn hardware on and off to save power.

You should now be able to measure the temperature in your home. If you’ve got a thermometer, you might want to confirm if this reading is accurate; if not, you might want to add a calibration constant to correct your Arduino reading.

Add the radio transmitter

We now have an Arduino temperature monitor set up. For it to be useful, we’re going to need to consider how we will access the data. Whilst an SD card adaptor might be an easier option, it would interrupt the measurements to have to keep removing and inserting the SD card periodically. In this tutorial we are, instead, going to

“ The Adafruit library saves us from having to write a program to deal with the communication protocol ”

broadcast our data using an AM radio to a receiver connected to another Arduino which will relay that data to computer over USB. If you have a Raspberry Pi, you could connect the receiver to that and skip the second Arduino entirely.

We’ve used the MX-FS-03V and MX-05V – a 434MHz transmitter and receiver pair that can be purchased online for pennies. With two separate pieces of spare wire soldered onto the transmitter and receiver and operating at 5V, communication was achieved at distances of 50m (not in line of sight). These should do the trick at short ranges within the home. If your walls are simply too thick and range becomes an issue, you may have to switch to Bluetooth or Wi-Fi instead.



Tutorial files available:
filesilo.co.uk

No wheel reinvention
With Arduino there’s often no need to go around trying to write a sketch to handle the communication protocol between two devices. There are dozens of libraries available for installation through the IDE for doing the tricky stuff. You can then spend your time trying to make a more useful device.

Every little helps...

For those of you determined to get every second out of your battery, someone has performed an investigation into how effective each of these power-saving measures is. Further info is given at <http://bit.ly/GammonPower>, including techniques such as reducing the internal clock speed and running your Arduino at voltages below 5V, providing some example sketches.

Connect the transmitter unit to the Arduino. You should be able to power the unit from the digital output pins as with the DHT22. Connect the data pin to a digital pin, too.

To interface with the radio, we're going to be using the VirtualWire library, which is available online but may be built-in on some versions of the IDE. VirtualWire is another wrapper library which handles the communication for many radio modules. All you have to do is designate the data pin and provide a message to send across, stored in a character array. To set up the radio transmitter, you'll need to set the high and low pins as before, initialise the library with the correct pin, and pick a bit rate. You may also need to invert the signal for some radio modules – the easiest way to find out is to play around.

```
vw_set_tx_pin(pinTXData);
vw_set_ptt_inverted(true);
vw_setup(4000);
```

You now need to form a message to send to the receiver. The plan is to take a temperature reading from the DHT22 (a floating-point variable) and convert it into a character array. This can be done easily on the Arduino using the double to string function: `dtostrf()`. VirtualWire will then take the character array and transmit it. This can all be done in a few lines in the main loop:

```
char message[10];
dtostrf(dataToSend, 6, 2, message);
vw_send(msg, strlen(msg));
vw_wait_tx();
```

Receiving messages on an Arduino is a very similar process. In setup, the reference to `tx` (transmit) becomes `rx` (receive) and in loop, `send` becomes `get_message`. The received message is then stored in a buffer of unsigned integers.

In a separate sketch, make a program that will attempt to receive data over radio indefinitely and upload it to a second Arduino. You can connect the receiver in the same way as the transmitter and let VirtualWire know which digital pin is for data. This process should be fairly straightforward, but both sketches have been uploaded to the cover disc. You should be able to iterate over the message buffer letter by letter, converting from integer to character, and print the buffer to serial:

```
if (vw_get_message(buf, &buflen)) {
    for (int i = 0; i < buflen; i++) {
        char c = buf[i];
        Serial.print(c);
    }
}
```

Log the data

Now we have received the message, we can relay it to a computer so that we can store the data and analyse it day-by-day rather than just collecting it from an SD card after a few weeks or months.

The easiest way to log the data from the Arduino is probably using a Python script. In the previous tutorial (LU&D 185) we covered sending messages over serial using Python from a Raspberry Pi. We're going to do much the same thing, but backwards. Then we can save it to file, with the program running continuously in the background just waiting for a message to come through.

Open a new Python script. To begin with you're only going to need to import the 'serial' library so that we can interface with the Arduino. In the Arduino IDE you should be able to view the name of the port that the Arduino is connected to under Tools > Port. In the Python script, you can then write:

```
arduino = serial.Serial(portName)
```

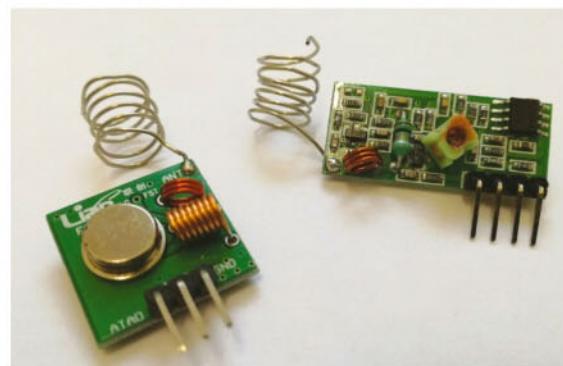
...to initialise the connection with the Arduino. We then simply open a new text file with the `open()` command and, until we exit the program, can read in characters from the serial buffer and write them to that file.

```
dataFile = open(myNewFile "w+")
while True:
    b = arduino.read().decode("utf-8")
    dataFile.write(b)
```

You may want to reformat the incoming data to make it easier to read. With the 'datetime' library, it's easy to create a string containing the current system time, which can then be added to the end of each line in your text file.

“ The easiest way to log the data from the Arduino is probably by using a Python script ”

After running both of your Arduinos and collecting data on your computing by running the script, you should have accumulated some interesting data about the temperature in your home. With Python's matplotlib library (or your favourite graphing program), you should



Above Radio transmitter and receiver pair; we've soldered a piece of wire to each to improve the range

be able to plot a scatter graph of your temperature data against time.

Read the data sheet

By this point you've already made something incredibly useful. However, we're going to take it one step further and make your Arduino as energy-efficient as possible. The benefits are obvious: by using the microcontroller chip's low-power mode and turning off pieces of hardware, we can not only save a few pennies in electricity, but also run the device off of a battery for much longer. With the right setup, it is possible to create something which can monitor and report the temperature that can run for months without you needing to interfere.

The following should work for the ATmega328 and 328P, which is available on many Arduino boards. You should familiarise yourself with the data sheet for your chip before proceeding – it is important that you are aware of how the Arduino can be 'woken up' after it is 'put to sleep' and how this affects your ability to interface with the board.

To begin, you will need to include three header files which let you control the chip's hardware:

```
#include <avr/sleep.h>
#include <avr/power.h>
#include <avr/wdt.h>
```

These allow you to put the microcontroller chip into one of the many sleep modes, turn off the hardware and interface with the watchdog timer – which we'll use to turn the Arduino back on again.

To begin, start by turning off the wasteful hardware. You won't need items such as the analogue-to-digital converter, the counters, the serial interface, even the USB itself. In loop, after interfacing with the sensor and the radio, turn off the wasteful hardware by writing:

```
ADCSRA = 0;
power_all_disable();
```

Set an alarm and go to sleep

For the Arduino to wake up, we'll need to set the watchdog timer – one of the few pieces of hardware that doesn't turn off. This interrupts the Arduino after a set number of clock cycles. First we allow changes to be written to the watchdog register and then, referring to the data sheet, we set the length of time the watchdog should wait before waking the Arduino (in this case eight seconds), we then reset the timer to zero:

```
WDTCR = bit (WDCE) | bit (WDE);
WDTCR = bit (WDIE) | bit (WDP3) | bit
(WDP0);
wdt_reset();
```

We've now cut off the hardware and asked for a wake-up-call. It's time to send the Arduino to sleep. First we select the sleep settings: 'SLEEP_MODE_PWR_DOWN' is the deepest sleep mode, with the largest savings. Sleep



Above The final device, assembled and placed in an old jar

mode is subsequently armed, executed and disarmed after waking.

```
set_sleep_mode(SLEEP_MODE_PWR_DOWN);
sleep_enable();
sleep_cpu ();
sleep_disable();
```

At this point you won't be able to do much to the Arduino. You'll just have to wait for it to wake up again. It's worth adding a large delay while testing, in case you need to upload something to the Arduino. Expect to spend a dozen minutes or so pressing the reset button and trying to find a window in which you can upload a new sketch before it falls asleep again. Be careful here.

Upon waking, the ATmega will execute a system reset unless an alternative function is defined. You can do this in a few lines at the top of the code. We'll just use it to turn off the watchdog timer, so that the rest of our sketch runs nicely. The Arduino will then carry on where it left off before going to sleep.

```
ISR (WDT_vect) {
    wdt_disable();
}
```

A line-by-line explanation of how to initiate power saving is available in the example sketch provided on the cover disc. It also comes with a few extra safety measures. A full description of the energy savings is available at www.gammon.com.au/power.

These measures should allow you to get the most life out any battery powering your Arduino project. If you've gone as far as to use a 'bare-bones' board, it's possible to get years of operation out of a relatively small battery. Having followed the tutorial this far, you'll now have a pair of devices which are able to measure the temperature, broadcast that over radio and store the data over the coming winter. Now, without guilt, feel free to turn your radiators on. ■



John Gowers

is a university tutor in Programming and Computer Science, with a strong focus on Java. He likes to install Linux on every device he can get his hands on.

Resources

■ OpenJDK 1.8

See your package manager or download from openjdk.java.net

■ JavaFX 8

See your package manager or download from openjdk.java.net/projects/openjfx

■ Eclipse IDE

See your package manager or download from eclipse.org

■ Spring Framework

Installation instructions in article

PART FIVE

Making a game in Java: Dependency injection

The Spring Framework is one of the most popular Java tools, and this project is a great opportunity to learn more about it



This month's Java tutorial has a slightly different focus from previous issues. Rather than make any substantive modifications to the functionality of our code, we will be rewiring our existing code in order to fit it into a Java software framework called Spring. Unlike traditional programming, where we take responsibility for creating an entire program, when we work with a software framework, our focus is on providing components for the framework and configuring it to use them.

The way we have set up our program, with important classes backed by interfaces, makes it well suited to this style of programming. We will be using the Spring Framework, which is probably the most popular framework for Java. Using Spring will allow us to incorporate an important design pattern – dependency injection – which cannot be used with Java alone.

If you haven't been following the series so far, you can get up to speed with a working version of the assignment from last month. Find the file **eggs.tar** on the cover disc, and download it to your system. In Eclipse, select General

> Existing Projects into Workspace, and then click 'Select archive file', followed by 'Browse...'. Navigate to **eggs.tar** and click 'Finish' to import the project.

Dependency injection

The key concept with dependency injection is that of a *dependency* between classes. A class **Master** has a dependency on a class **Service** if the class **Service** is mentioned somewhere in the code of the class **Master**: for example, if **Master** has a field of type **Service**. Rather than have an explicit dependency between the two classes, it is better practice to write an interface **ServiceType**, implemented by the **Service** class, and make the **Master** class have a dependency on this interface, rather than on the implementing class. The **ServiceType** interface should provide all the methods from **Service** that are used by the **Master** class. This decouples our program and allows us to use different implementations of the **ServiceType** interface, which are passed into the **Master** class through its constructor.

This approach is the one we've taken in the Eggs game from the beginning. All the service classes – `TextMap`, `ConsoleOutputViewer`, `ServerHead` and so on – are backed by interfaces, and classes such as `StandardGameModel` have a dependency on these interfaces, rather than on the implementing classes. The interfaces provide the methods that the `StandardGameModel` uses to function, but their implementations can have quite different functions: for example, the classes `ConsoleOutputViewer` and `ServerHead` both implement the `OutputViewer` interface; by replacing one with the other, we converted the standalone program into a game server that client programs could connect to.

All this came at a small cost, however. In the main methods for our `App` or `EggsServer` classes, we needed to instantiate every service class used in our program, passing objects in to other objects as dependencies. Our program is simple enough, but for a more complicated program, keeping track of all the dependencies could turn out to be a major task. Moreover, if we wish to use our classes to create multiple front-ends that have some dependencies in common and some that are different, we'll need to create helper methods to fill in the common dependencies if we want to avoid repeated code.

A way to get round this problem is to use *dependency injection*. This is a pattern whereby a class's dependencies are filled in automatically by means of a configuration file or Java annotations. With dependency injection, we can specify which

“ Dependency injection would ideally be provided within Java itself, but since it is not, we have to use an external framework ”

versions of each dependency we wish to use, and then have these dependencies 'injected' into the classes that need them at runtime. For example, instead of having to pass an `OutputViewer` object into the `StandardGameModel` class directly, we can register that we wish to use the `ConsoleOutputViewer`, and let dependency injection handle inserting this particular dependency into the class when it is required. This means that we do not need to keep track of the inner workings of the `StandardGameModel` class within our main class.

There are two main dependency injection frameworks: Spring and Google Guice. As the README for the Google Guice project says, dependency injection would ideally be provided within Java itself, but since it is not, we have to make do with an external framework instead. In this project, we will be exploring the Spring Framework, but you might like to look at Google Guice as well.

Dependency	A class or interface used by another class
IoC Container	The part of the framework that is responsible for running the program and injecting dependencies
Bean (@Bean)	An object that the IoC container can use to fulfil a dependency
Service (@Service)	A class that can be instantiated to form a bean
Bean Factory	Object provided by Spring that creates beans
@Autowired	This method will be called when a bean is instantiated. The parameters of an autowired method are Spring dependencies
@Scope	Used to decide whether a bean will be instantiated only once ("singleton") or every time it is needed ("prototype")

Spring concepts

The core concept in Spring is that of a *spring bean*. Put simply, a bean is an object that is created and controlled by the Spring Framework. Rather than construct the object using `new`, we can ask Spring to provide us with an instance of that particular bean. If the bean's constructor takes parameters, Spring will search for additional beans that it can use to fill them in. For example, in our program, we can tell Spring to create a bean of type `GameLogic` and it will create an instance of the `StandardGameLogic` classes, as well as instances of all the classes that need to be passed into that class's constructor.

There are two ways to register beans with the Spring Framework. The older way is to write an XML file containing data which Spring will read in order to register the bean classes. The newer way, which we shall be using in this issue, is by using Java annotations.

If we precede a class declaration with the annotation `@Component` or `@Service`, Spring will create an instance of that class whenever it needs a bean of that particular type. For instance, if one of our beans has a dependency on the `OutputViewer` interface, and we have marked the `ServerHead` class with the `@Service` annotation, then Spring will instantiate the `ServerHead` class in order to satisfy the dependencies of the first bean.

The second annotation we can use to create beans is the `@Bean` annotation, which we place immediately before a public method definition. For example, if we write a method returning a `Player` object, and add the `@Bean` annotation to that method, then Spring will call that method and use the return value whenever it needs to satisfy a bean dependency of type `Player`. If the method takes in parameters, then Spring will look for more beans that it can insert in order to call that method.

In order to construct beans, we use a device called a Bean Factory, which is provided by Spring.

There are several things that can go wrong in this process. For example, if Spring tries to create a bean of

Above Spring cheat-sheet. Spring adds a number of key concepts to the basic Java paradigm in order to model dependency injection

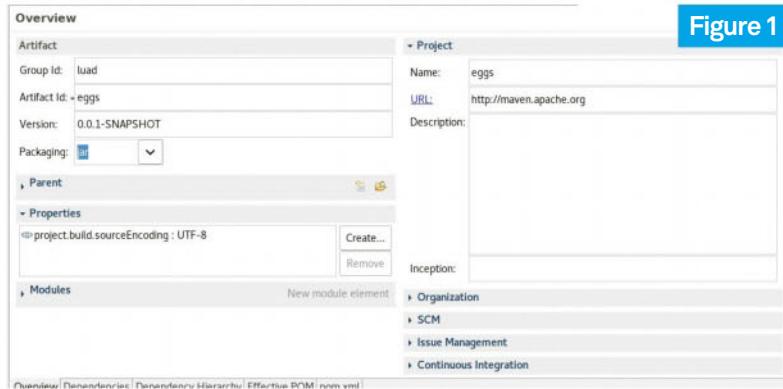


Figure 1

Above Eclipse provides a useful graphical interface for editing the Maven configuration file `pom.xml`

Right In order to import a Maven dependency, we give its name and group ID, and Maven will download the JAR files from its repositories

type `GameModel` and finds that none has been registered in the project, then it will terminate the program with an error message:

A component required a bean of type 'lعاد.eggs.GameModel' that could not be found.

The other problem occurs when we register two beans of the same type in the same program. Then Spring will give us an error message that looks a bit like this:

expected single matching bean but found 2

In this case, we have to use further annotations in order to tell Spring which beans to use where. We will not cover this in this issue, but you might like to look up the `@Primary` and `Qualifier` annotations.

Installing Spring

The first step we need to take is to install the Spring Framework dependencies into our Eclipse project, which we will do using Maven. The project we have provided

Java annotations

If you've never used any kind of Java framework before, you might not have met many, if any, Java annotations. Annotations are used to give the compiler extra info about a particular class, object, method or field. For example, if we mark a method with `@Override` then the compiler will refuse to compile the module if the method does not in fact override some supertype method.

Annotations are more powerful than this simple example suggests. Using the Reflection API, a software framework such as Spring can query whether or not a method or class has a particular annotation when it is deciding how to handle it. Annotations may also take 'elements': string data fields which can further inform the framework's behaviour.

Just like interfaces and classes, annotations are declared in their own files and we have to import them if we wish to use them in another package. Importing annotations in Eclipse is as easy as importing classes or interfaces. If we type the annotation `@Service` above one of our classes, Eclipse will put an error mark by the side of our code (see Figure 3). If we then click on this error mark, Eclipse will give us the option to import the annotation from the package `org.springframework.stereotype` (assuming that we have our Spring dependencies installed already). If we double-click on this option, then Eclipse will automatically insert the appropriate `import` statement into our code.

is a Maven project in Eclipse, but we have not used the Maven functionality yet so far.

Note: in this section the word 'dependency' will refer to an external Java software package which we can add to our project. Everywhere else, we will use it in the sense of dependency injection, as above.

If we expand the directory of the 'eggs' project in Eclipse, we see a file called `pom.xml` at the bottom. Double-clicking on this file brings up the dialog shown in Figure 1. The `pom.xml` file is responsible for configuring the Maven build. Eclipse provides this useful graphical dialog which we can use to edit `pom.xml`.

The first change we want to make is to make our project a child of the Spring Framework project. This will make it easier for us to add dependencies without having to specify their version numbers. On the Overview tab, expand the section titled Parent and type

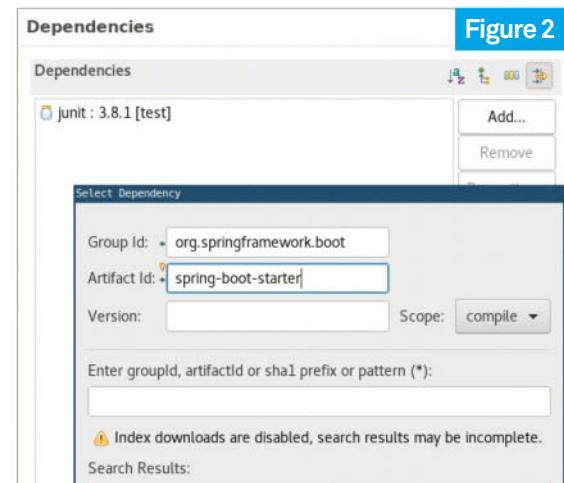


Figure 2

`org.springframework.boot` into the Group Id field, `spring-boot-starter-parent` into the Artifact Id field and `1.5.8.RELEASE` into the Version field.

For the next step, click the Dependencies tab at the bottom, which will bring up the dialog in Figure 2. We need to add two dependencies: Spring boot starter and Spring context. To do these we click the 'Add...' button. For Spring boot starter, type `org.springframework.boot` in Group Id, and `spring-boot-starter` in Artifact Id and hit 'OK'. For the second dependency, Spring context, the Group Id is `org.springframework` and the Artifact Id is `spring-context`. We don't need to include a Version field, as this is taken care of by the Parent artifact.

Last, we need to run the Maven build to install these dependencies. In Eclipse, select Run > 'Run configurations...' to bring up the dialog shown in Figure 4.

Click on Maven Build at the left, then press the New icon at the top. Under Name at the top, give the new configuration a name; for example, 'eggs install'.

Underneath, click the button 'Workspace...' under the 'Base directory' text field, then click on the name of the 'eggs' project and press **Enter**. Under Goals, further down, type `install`. Then, click 'Run' at the bottom to start installation of dependencies.

The Maven build will now download the required JAR files for the Spring Framework from the internet, if they are not on your computer already. You can check that they have been downloaded by expanding the Maven Dependencies folder for your package in the Package Explorer pane of Eclipse. If it is full of JAR files related to Spring, as in **Figure 5**, then we are ready to get started.

Getting started with dependency injection

Create a new package in the project called `luad.eggs.network.server.springboot`, and create two classes in it: `EggsServer`, which should contain a `main` method, and `EggsServerRunner`, which should implement the `CommandLineRunner` interface from `org.springframework.boot`. The `main` method in the `EggsServer` class is one line long:

```
public static void main(String[] args)
{
    SpringApplication.run(EggsServer.class,
args);
}
```

We also need to add the following annotation immediately above the line `public class EggsServer` at the start of the class definition:

```
@SpringBootApplication(scanBasePackages = {
    "luad.eggs"
})
```

From a conventional Java programming point of view, this looks odd: our main method hasn't mentioned any of our other classes at all. What the `SpringApplication.run` method does is to search for a bean implementing the `ApplicationRunner` interface or the `CommandLineRunner` interface and to use the `run` method of that bean to run the program. So our code to start the application will go in the `run` method of the `=EggsServerRunner` class.

The annotation `@SpringBootApplication` tells Spring that the `EggsServer` class is the entry-point to our Spring application. Then the `scanBasePackages` element tells Spring that it should search for beans within the `luad.eggs` package and all subpackages.

Since we want Spring to recognise our `EggsRunner` class as a bean, we must also add the `@Component` annotation immediately above the declaration `public class EggsServerRunner`.

```
4 import java.util.Observable;
5 import java.util.Set;
6
7 import luad.eggs.mapTiles.MapTile;
8
9 /**
10  * A model of the game using text-based commands.
11 */
12 @Service
13 P - Import 'Service' (org.springframework.stereotype)
14 { @ Create annotation 'Service'
15   ↗ Change to 'ServiceMode' (javax.xml.ws)
16   ↗ Rename in file (Ctrl+2 R)
17   ↗ Fix project setup...
```

Figure 3

Create, manage, and run configurations

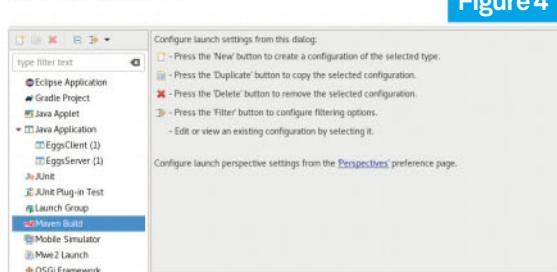


Figure 4

Above In order to run a Maven project, we must specify 'goals' for Maven to aim for

If you haven't already, create the method `public void run(String... args)` inside the `EggsServerRunner`, and put a simple print statement inside it (e.g., `System.out.println("Hello, world!");`), so that we can verify that the framework is working.

Go back to the 'Run configurations' dialog and create a new Maven build, called 'eggs run'. As before, select the 'eggs' project under 'Base directory', but now type the following under Goals:

```
spring-boot:run
```

When you run the configuration, you should see output as in **Figure 6**. Spring produces a lot of output of its own, but among it, the program has printed out `Hello, world!`.

Setting up our beans

We now want to mimic what the original `EggsServer` class does in the Spring Framework, and in order to do that we need to make some more beans for Spring to use. If we look at the loop in the original class, we see that the first main task it has is to create a `GameModel` object. It does this by creating an instance of the `StandardGameModel` class, passing in `Player`, `Map` and `OutputViewer` objects that it has created before.

Since we are using dependency injection, we don't want the `EggsServer` or `EggsServerRunner` classes to have to define the particular implementations of the `GameModel`, `Player`, `Map` and `OutputViewer` objects that we are going to use. Instead, we want to be able to tell the `EggsServerRunner` class to create a `GameModel` object for each client connection, and then let Spring handle the details of how to set those objects up.

For Spring to do its work, we first need to let it know which classes it is allowed to use as beans. In order to do this, we go into the `StandardGameModel`, `HumanPlayer`, `ServerHead` and `StreamInputController` classes and add the following annotations immediately above the class declaration (i.e., the line `public class [...]`):

```
@Service
@Scope("prototype")
```

The first line tells Spring that it can use this class as a bean to satisfy dependencies. For example, if Spring is trying to create an object of type `OutputViewer`,

■ XML for dependency injection

You might like to try using the older method of dependency injection with Spring, which uses a configuration file written in XML instead of annotations to specify the locations of the beans and their dependencies. There are some advantages to this method, which we have not covered in the main article. For example, it allows you to keep your beans and dependencies entirely separate from your code, meaning that you could effectively change the entire behaviour of your program by swapping out one XML file for another. See the Spring documentation at <https://spring.io/docs> for more information.

Left Importing new annotations in Eclipse is just as easy as importing new classes or interfaces

More magic with Spring
 Spring is far more than a dependency injection framework. One useful tool is the `@Async` annotation, which specifies that a method should be performed *asynchronously*; i.e., in its own thread. One advantage of this over new `Thread(runnable).start();` is that the method can take parameters. In order to use `@Async`, we must annotate our main class with the `@EnableAsync` method. We must also provide a bean of type `TaskExecutor` – this is an interface provided by Spring that represents an object that holds a pool of threads and can use them to run different methods asynchronously.

Below We can check that our Maven dependencies have installed by looking inside the Package Explorer in Eclipse

it will create a `ServerHead` object in order to satisfy that dependency.

The second line, `@Scope("prototype")`, is equally important. By default, Spring beans are *singletons*: that is, Spring will create the beans it needs when it boots up and will then use the same instances of those beans to satisfy dependencies whenever it needs them. This works well when the beans take the role of services that the entire program can use, such as in the single-player game. However, in the multiplayer game, it is important that we create separate `HumanPlayer` and `ServerHead` instances each time we need one. For that reason, we tell Spring to use *prototype* rather than singleton scoping: now Spring will create a new bean of a given type each time it's needed and the bean will live for the duration of the life cycle of the object that depends on it.

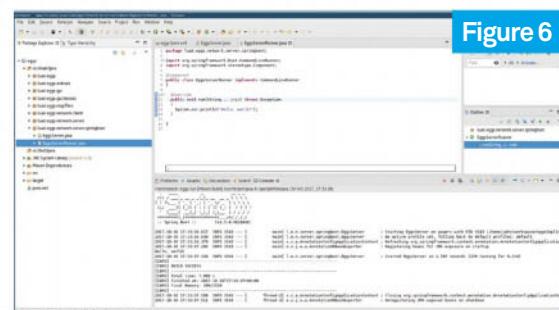
Dependency injection enables us to fully abstract out the networking part of the code to another class

The game map, on the other hand, *should* be a singleton: we want to create a single instance of the map and have all instances of the game model use it. So we add the annotation `@Service` to the `Map` class, but we do not add `@Scope("prototype")` after it.

@Autowired and @Bean methods

There are a couple of steps that need to take place after we create the basic game objects and before we run the game. When we create the game map, we need to call the `readMap()` method to read in the right map file. And when we create a new player, we need to set its position.

So far, when we have marked a class as `@Service`, Spring has only called the constructor of the class when it instantiates it as a bean. Now, we want it to call additional methods too. To do this, we mark the methods we want Spring to call with the `@Autowired` annotation. In our case, we add the annotation `@Autowired` immediately before the declarations of the `readMap()` method from



Above Spring prints out a lot of its own output to the console, but we can still find our own text output if we look.

the `TextMap` class and the `setPosition()` method from the `HumanPlayer` class. Both these methods take parameters: the filename of the map and the point to start on. So we open the `EggServer` class and add new bean methods to return this information. For example, you might create a method that looks like this:

```
public String getMapFileName()
{
    return "src/main/resources/maps/bigMap.txt";
}
```

...and a similar map `getStartingPosition()` returning an appropriate `Point2D` object. To tell Spring it can use these methods to create beans, we add the `@Bean` annotation immediately before these method declarations.

Setting up networking

Thanks to the magic of dependency injection, we can fully abstract out the networking part of the code to another class. Create a new class called `ClientConnection` in the `luad.eggs.networking.server.springboot` package. This goal of this class will be to provide a bean for the `ServerHead` class to use in its constructor every time a client makes a connection.

Create a field in the class of type `ServerSocket` called `serverSocket` and write a constructor that sets this field by taking in a port number as a parameter and passing it to the `ServerSocket` constructor. Then create a method in the `EggServer` class, marked with the `@Bean` annotation, that returns the port number `9009`. The `ClientConnection` class will use this bean to populate the `serverSocket` field when it is being set up.

Create a second field called `clientSocket`, of type `Socket`; create a method `fetchNewClientConnection()` that sets this field using the `serverSocket.accept()` method. The `fetchNewClientConnection` method should return a Boolean value: `true` if the connection was made successfully (i.e., `clientSocket != null`) and `false` otherwise.

Now we are in a position to create a bean that will return a `Socket` object. Create a method in the `ClientConnection` class called `getClientSocket()` that returns the value of the `clientSocket` field. Mark this field with the `@Bean` and `@Scope("prototype")` methods so that it will try to return a new value each time.

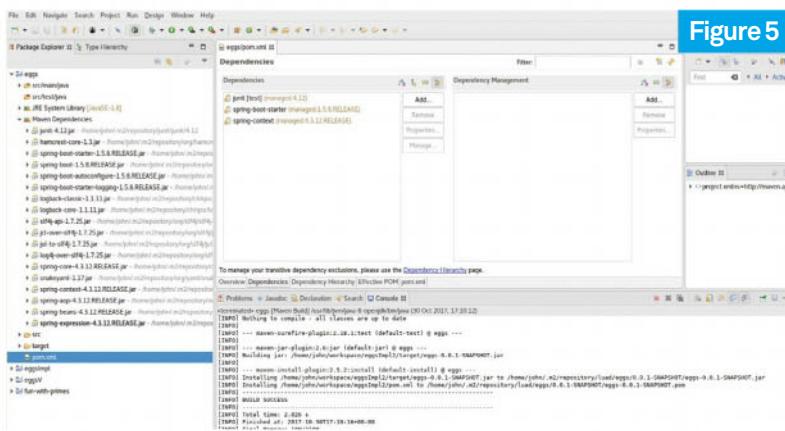


Figure 6

Figure 5

Under the hood

Dependency injection in Spring works using reflection: a set of tools that allows us to inspect low-level features of the program, such as the classes and methods it contains. Using reflection is usually considered a bad idea, since it can make programs hard to debug. However, it also allows us to do things that we would be unable to do otherwise, such as dependency injection.

The Java Reflection API allows us to find all the classes in a package and its subpackages and iterate over all the methods in these classes. (This is what the `scanBasePackages` annotation element is used for.) These methods are modelled as objects of the `SpecialMethod` class that is part of the Reflection API. Spring can then call a method `getAnnotation()` to inspect the annotations of each class and method. For example, if Spring is looking for bean methods, it can call `getAnnotation("Bean")` on each method and check to see if the value returned is null. If it is not, then Spring can call the `getParameterTypes()` and `getReturnType()` from the Reflection API to find out what type of bean the method provides and what it needs to be invoked.

Reflection can be computationally expensive, so Spring tries to do as little of it as possible. This means creating a cache at the start of the program that holds all the relevant information about which classes, fields and methods have which annotations.

The `ServerHead` bean will use this `Socket` bean to set up its connection when Spring instantiates it. To use the same bean to set up the `StreamInputController` bean, we will need to create a new bean inside `EggsServer` that takes in a socket as a parameter and returns its input stream. Mark this bean with `@Scope("prototype")` as well. Once we have written this bean method, it will use the socket bean from the `ClientConnection` class to set up the input stream for the server.

Since the `ClientConnection` class itself has singleton scope, Spring will not create a new instance of it each time, instead using the same instance. This means that we will only get a new client connection when we call the `fetchNewClientConnection()` method. To avoid an error, it's important that we call this method before trying to instantiate any bean that has a `Socket` dependency.

Create the game loop

Now we are in a position to create the main game loop. We now want to tell Spring to attempt to create beans of type `GameModel` and `StreamInputController` each time a client connects.

So far, we have seen how to tell beans to create other beans in order to set themselves up, but we have not yet seen how to start up a bean in the first place. In order to do this, we need to use a Bean Factory, which is a special object provided by Spring for the purposes of instantiating beans.

Creating a Bean Factory is quite easy. In fact, all we have to do is add the following field declaration to the `EggsServerRunner` class:

```
@Autowired  
private BeanFactory beanFactory;
```

Since `BeanFactory` is part of the Spring Framework, the `@Autowired` annotation means that Spring will

automatically populate this field with a `BeanFactory` bean provided by Spring itself. We can now create, for example, a `GameModel` bean using the code:

```
GameModel gameModel = beanFactory.  
getBean(GameModel.class);
```

In order to complete the program, remove the print statement from the `run()` method of `EggsServerRunner` and replace it with the code to make the game work. First, we need to use our Bean Factory to instantiate a bean of type `ClientConnection`, which we will use for the networking part of the code. After this, write a `while` loop that repeatedly calls the `fetchNewClientConnection()` method from this object and then instantiates two beans of type `GameModel` and `StreamOutputController` using the Bean Factory. Last, add the `GameModel` bean as an `Observer` of the `StreamOutputController` object, and call `new Thread(controller).start();` to start listening to input from the client.

If we look back at the original `EggsServer` class, we see that the main loop contains two lines of code that we have not yet accounted for:

```
map.addPlayer(player);
```

...and:

```
player.addObserver(gameModel);
```

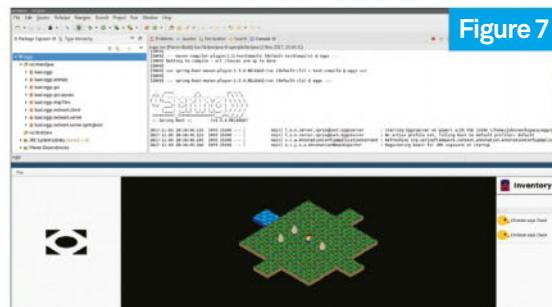


Figure 7

Left When we have the server running on the Spring Framework, clients should be able to connect and play the game as they could before

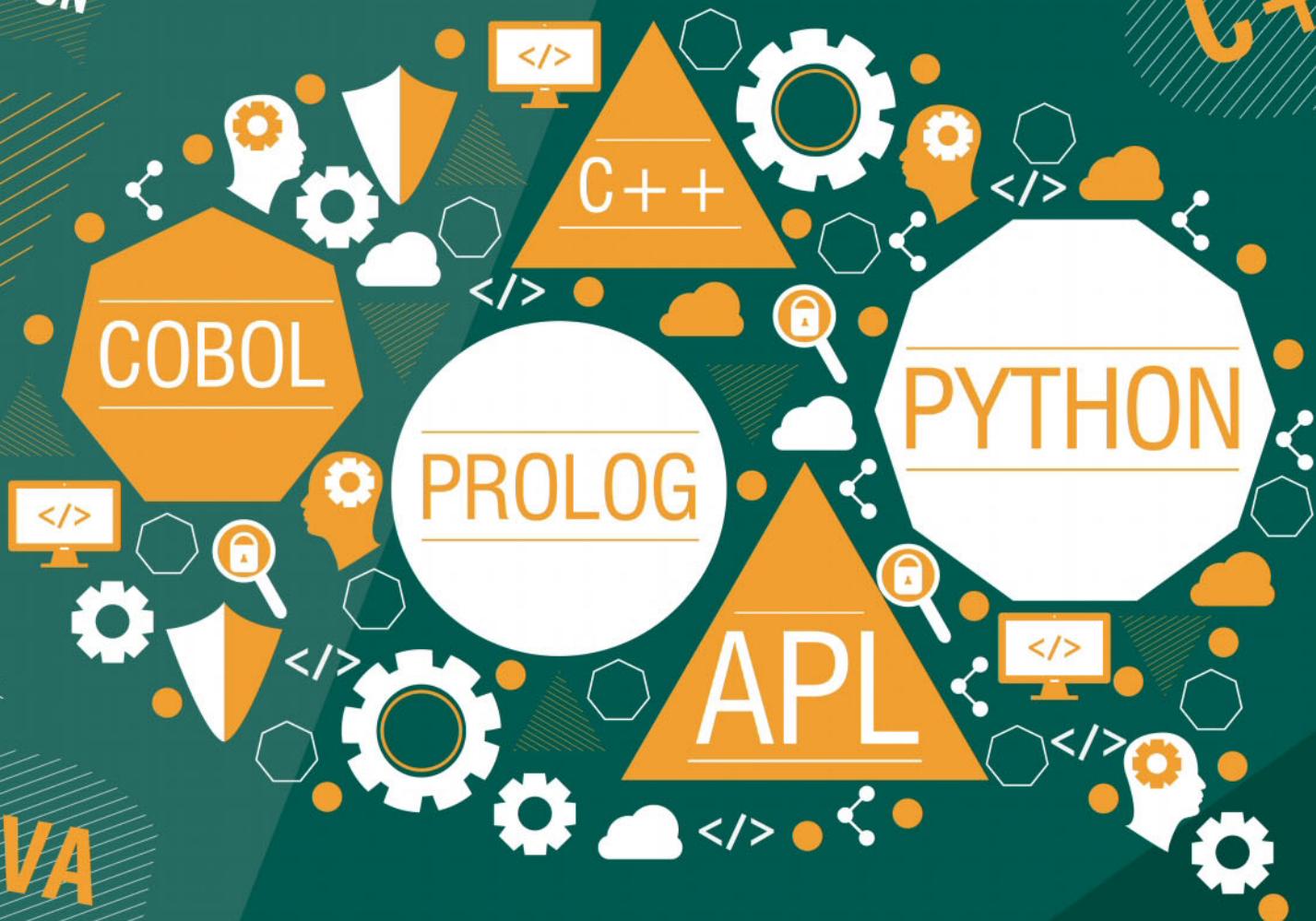
It's going to be a bit tricky to include these lines of code in our new, Spring-based version of the server. Instead, it makes sense to move them into the constructor of the `StandardGameModel` class. So we add the following two lines at the end of this constructor:

```
map.addPlayer(player);  
player.addObserver(this);
```

If you've done everything correctly, then the program should run exactly as it did before (see **Figure 7**). Run the server using the `eggs run` build configuration we created earlier and then fire up a couple of clients in the usual way. If it doesn't work, use the debugger and the Spring error output to try to find out where you went wrong. ■

PYTHON

C++ X



THE FUTURE OF PROGRAMMING LANGUAGES

Mike Bedford takes a tour of unusual and up-and-coming programming languages to investigate what the future may hold

AT A GLANCE

Where to find what you're looking for

- **Unusual languages, p62**

Many languages have much in common with so many others but a few have dared to be different. We investigate a handful of languages which break the mould and, in so doing, offer a very different way of programming. Each has something to offer the more adventurous programmer.

- **Languages to watch, p64**

The programming scene is in a state of constant change so it's extremely likely that we will be using some different languages in ten years' time. It's also likely these languages will be starting to gain momentum already, so we take a look at today's up-and-coming languages.

- **Future languages, p66**

To get a feel for what the future of programming might hold, we catalogue the development of languages over the past sixty odd years and build on these foundations by presenting the views of those who are involved in leading-edge research into programming languages.

A recent survey of programming languages suggests that programmers are all too predictable. Of those languages used by Linux coders, Python, C, C++ and Java have shared the top few slots for desktop applications for several years and, if we broaden our view to take in web applications, we find that JavaScript and PHP are also popular. With the odd exception, such as the importance of C# on the Windows platform, the list is broadly similar for other operating systems.

In one sense, this isn't too surprising. Considering career prospects, for example, it pays to learn a language that will maximise your employment opportunities

“Despite the obvious draw of the most popular languages, there are good reasons to look elsewhere”

and this tends to mean those that are most commonly used in business. There are also other good reasons to pick the most popular ones for personal use. Perhaps most importantly, the more prevalent a language is, the more likely you are to find adequate community support.

Despite the obvious draw of the most popular languages, there are also good reasons to look elsewhere. After all, if nobody had ever dared to push the

envelope, we'd still be using COBOL and Fortran. What's more, not everyone thinks in the same way, so it's quite possible that a less popular language could appeal to you and make you more productive.

So, if you're prepared to consider the alternatives, we're here to help by introducing you to some languages you may not have considered as well as thinking about what the future might hold. A grand tour of programming languages would be a serious undertaking – one list we discovered contained no fewer than 710 entries – so we're restricting ourselves to certain categories and, even then, being very selective. If we've managed to whet your appetite, therefore, we trust that you'll continue this journey of discovery yourself.

First of all we highlight a few languages, which, despite not being anywhere close to the top of the league tables, offer a very different way of working and have enthusiastic followings. These languages often tend to favour particular types of application in which they have undoubted strengths and so might not be for everyone, but if your needs match the features offered by one of them, it could be exactly what you need.

Even if you don't have a genuine requirement as a programmer you probably have a general interest in the wider

programming landscape so we trust you'll find this overview absorbing. We've referred to these as unusual languages, as indeed they are, although in most cases, that phrase isn't used in a disparaging sense.

Next, we identify the languages that some experts think may eventually topple Python, Java and C++ from the top spots – these we're calling the up-and-coming languages to watch. Of course, there's no guarantee that any of these will ever share the success of C++ or Java, and there's really no surefire way of predicting which will be the next must-have language. It often takes a long time for a language to receive recognition and gain critical mass.

Indeed, the past shows that a long history of lacklustre performance is no indication of a future of mediocrity. Python, for example, was first conceived in the late 1980s and launched in 1991, but it was 2006 before it gained a top-ten position in the TIOBE (The Importance of Being Earnest) Index of the popularity of programming languages. It took a full 26 years to reach its highest-ever position of number four that it enjoys today.

Finally, we'll investigate current research into programming languages, which, despite over 60 years of development for high-level languages alone, is still going strong. Researchers are developing better ways of instructing computers on what we need them to do and we ask a couple of experts to describe the current areas of interest and predict how soon these initiatives will yield a new way of working.

GO

Unusual languages

Languages that couldn't be more different from those commonly used today

Here we look at languages that can be considered unusual in the sense that they are not commonly used, but many are by no means new. But this doesn't necessarily mean that they're failed.

Unusual languages are often designed for niche applications or, perhaps, they appeal only to a minority of programmers but still attract enthusiastic support. It's entirely possible, therefore, that one of the languages discussed here could provide a solution for you or, at the very least, provide an interesting lesson into programming more generally.

Prolog

Prolog might date back to 1972 but, unlike most of the languages used today, which are third generation, Prolog has been described as a fifth-generation language in some quarters. This is because it is used to define the problem but doesn't tell the computer how to solve it in the sense of providing an algorithm. Working in Prolog involves defining relationships and rules and

“It can be used to solve any computable problem, just like Python or C++ **”**

then making queries about the database of these relationships and rules.

It soon became the language of choice for artificial and natural language applications – indeed, most introductions to the



Above When first introduced, APL needed a special keyboard and golfball printer to cope with its strange character set. Recent changes amended this, thanks to improved font technology and on-screen keyboards

language concentrate on AI. Despite the fact that this might suggest that Prolog is a language solely for niche applications, it is Turing Complete, which means that it can be used to solve any computable problem, just like Python or C++ – although realistically you probably wouldn't want to use it for mainstream applications.

A trivially simple example would be to provide the fact defined by `wine(chardonnay)`, which means that Chardonnay is a wine, then the rule `drink(X) :-`

`wine(X)`, which means that anything that's a wine is a drink. Now, if you were to issue the query `?- drink (chardonnay)`, meaning 'is Chardonnay a drink?', Prolog would respond with the answer 'true'.

Given that new programmers learning Java or Python write code to solve trivially simple problems, we trust that you'll recognise that much more interesting tasks can be solved in Prolog and very sophisticated expert systems can be put together this way. There's no shortage of Prolog implementations, many for Linux, but if you just want to get a feel for the language, a good option is to use the online version at <https://swish.swi-prolog.org>.

COBOL

COBOL is included here to illustrate how different some languages are – even ones intended for mainstream programming. Dating back to 1959, it was once the language of choice for business as opposed to scientific programming and it is still the 25th most popular language, but mostly for maintaining legacy systems. So it's unusual in the sense that it's almost never used for

TIMELINE

1842

1948

1949

1952

Ada Lovelace writes the first computer program, designed to run on Charles Babbage's unfinished Analytical Engine

The 'Small-Scale Experimental Machine' becomes the world's first stored program computer, programmed in machine code

EDSAC is programmed in one of the first assembly languages, Initial Orders. The assembler was hard-wired into it

Autocode is the first high-level language. It's the ancestor of all today's languages, even though it never took the world by storm

new applications today, although there are still employment opportunities for those proficient in this language of yesteryear.

The language is very powerful in the way it handles hierarchical data structures, which is very useful in traditional data processing tasks, but its most unusual aspect is its instructions. Called sentences, they have full stops at the end, but this isn't the limit to its similarity with the English language.

The idea was that accountants and others with no expertise in programming could look at a program and understand its function. So, where most languages might use a statement like `Gross_price = Nett_price * Vat_rate`, in COBOL this would be `MULTIPLY Nett_price BY Vat-`

QUICK GUIDE

Esoteric languages

This type of language is never used to solve real world problem but you might find them intriguing, nevertheless. Some were designed as a challenge to programmers because they are so difficult to use. Java2K, for example, uses base 11 arithmetic, with space to represent the digit 10, so you can't make a program easier to read by adding spaces. Furthermore, it's a probabilistic language rather than a deterministic one (don't ask). Others, like INTERCAL, are more of a joke. It has no GOTO type instruction but it does have COME FROM, which takes some getting used to.

Slightly less perversely, but not a lot, are those esoteric languages designed to be minimalistic so require very simple compilers. Brainf*** (yes, it's a swear word) is a universal language but it has just eight instructions, each represented by a single character. One compiler is just 100 bytes long.

“Unusual languages are often designed for niche applications or, perhaps, they appeal only to a minority of programmers”

rate GIVING Gross-price. GnuCOBOL and OpenCobolIDE are a couple of open source COBOL projects, should you want to have a go at this unusual language.

APL

APL stands for A Programming Language and was first released by IBM in 1964. It can be used to write a program in the sense of a sequence of instructions which is then executed, but it can also be used as a glorified calculator, entering instructions to which the result is immediately displayed. This interactive programming environment is one of the main features that draws people to APL, even today.

Also unusual is the fact that a lot of computations that would take several lines of code in most languages require just a single statement in APL, e.g. If you were to type `8!`, APL would respond with the answer 40320, which is 8 factorial. Matrix manipulation can also be carried out using single instructions.

APL uses single characters for many of its operations and this terse nature is both a blessing and a curse. It's an advantage in that you don't need to type a lot to get an answer but it's notoriously difficult to learn, let alone maintain code. Not only that but, because the ASCII character set doesn't contain nearly enough symbols, a unique character set is used. At one time this required a special keyboard, or at least a good memory of which combination of keys to use to access a given character, but many of today's implementations overcome this by offering on-screen keyboards.

To give you a feel for both the power and the seemingly unfathomable nature of APL,

the following statement causes all prime numbers up to the value R to be displayed:

```
(~RεR○.×R)/R+1↓iR
```

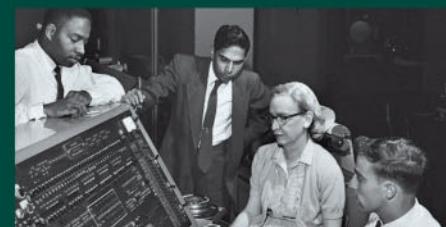
If this has inspired you to try it out yourself, and perhaps take a degree of satisfaction in mastering such an arcane language, <http://tryapl.org> is a good online APL implementation. Alternatively, there are several versions available to install on your PC, some of which run under Linux.

If our look at these three unusual languages has whetted your appetite to delve into a few others that will never appear at the top of league tables, give Lisp a try – and, in particular, Common Lisp – and J, which was inspired by APL but doesn't use its weird character set. ▶

```
APL provides an interactive programming environment
A Store 10,000 dice throws
throws←1000000
A Display result
throws
6 3 1 6 1 5 2 4 2 4 6 4 4 4 4 4 3 2 1 3 6 2 1 4 2 5 4 2 3 6
A How many ones were thrown?
+/1-throws
1630

A Show Frequency of all six possibilities
+/((6)○.×throws
1630 1661 1699 1700 1707 1603
```

Above APL is able to provide an interactive programming environment



Above COBOL, which was developed by Grace Hopper, looks odd by today's standards

CC BY 2.0 Jan Arkester/jn

1957

1958

1958

1959

Fortran is the first mainstream, high-level language, ideal for scientific application – the main use of early computers

LISP uses the paradigm of list processing and gains support for AI, it's different in nature from other early high-level languages

ALGOL, the first block-structure language, is designed and encourages improved programming techniques.

COBOL is launched for business applications, setting the scene for different languages for different applications.

Languages to watch

The ones that could topple Java, C, C++ and Python

Languages come and go and, if we were to analyse the most used ones over the years, we'd surely come up with a different list for each decade. While still in the top ten, for example, in the last 10 years PHP use has plummeted from a usage figure of over ten per cent to less than two per cent. Conversely, Python has increased from just over one per cent in 2002 to almost five per cent today. This begs the question of what we'll be using in another decade or so.

Languages rarely take off in a big way overnight so, almost certainly, the ones that we'll be using in ten years' time will already have been released and will already be starting to gain popularity. Accordingly, we're looking here at what we've referred

to as the up-and-coming languages of the programming world.

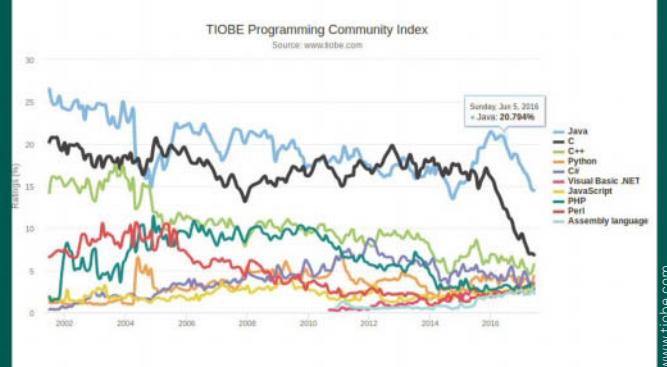
Coming up with a list of the top languages to watch, would involve making a very subjective decision. Rather than relying on our own judgement, therefore, we've distilled the views of several commentators to come up with the top few.

We've looked to the developers of two that represent quite different advanced programming concepts to see what they believe sets their creations apart from the competition. We also mention a few others that would be worthy of consideration if you're looking for the next big thing.

Go

Perhaps the language that most people are getting excited about, Go is an open-source project developed by a team at Google and elsewhere. Unlike many up-and-coming languages it's comparatively new, first appearing as recently as 2012. The team say that it's designed to make programmers more productive – so nothing new there, then – but in particular they make the following claim:

"Go is expressive, concise, clean, and efficient. Its concurrency mechanisms make it easy to write programs that get the most out of multicore and networked machines, while its novel type system



Above The TIOBE index: Tracking the rising stars of the programming world

enables flexible and modular program construction. Go compiles quickly to machine code yet has the convenience of garbage collection and the power of run-time reflection. It's a fast, statically typed, compiled language that feels like a dynamically typed, interpreted language."

You can try Go for yourself online at <https://play.golang.org> but there are also lots of online educational resources which would be well worth investigating if you're new to the language. Take a look at the online tour at <https://tour.golang.org>. Then, once you've exhausted the online resources, you'll probably want to install Go on your PC. For that, you should visit <https://golang.org/doc/install>.

Haskell

Haskell is another open-source project boasting over 20 years of development. According to its developers, it is "a polymorphically statically typed, lazy, purely functional language, quite different

BACK IN TIME

Programming the Turing Machine

Before real-world digital computers, Alan Turing came up with a hypothetical machine capable of universal computation. It was never designed as to be practical and it wouldn't have been fast, but the concept was instrumental in developing the theory of computation.

Despite it not being a physical computer, it has been simulated and this provides the opportunity for anyone to get a feel for what programming it would be like by heading over to www.turingmachinesimulator.com for a tutorial and an online simulator.

TIMELINE

1964

BASIC is an easy-to-use first language, designed for beginners. It is used extensively on 1980s home computers

1964

APL offers an interactive programming environment. Using a unique character set, it can express problems concisely

1972

Smalltalk is one of the first languages to feature object orientation. It influenced many of today's popular languages

1972

C is launched. Building on ALGOL, it becomes very popular, eventually giving rise to derivatives like C++ and C#

“A list of the new languages that are attracting attention would be a long list indeed”

from most other programming languages.” They further suggest that even if you are not in a position to use Haskell in your programming projects, learning it can actually make you a better programmer in any language.

Since the functional aspect is, perhaps, what the community seem to suggest is Haskell’s key feature, it would be worthwhile looking in a bit more detail about what this programming paradigm involves. Again, quoting its developers, this sets it apart from languages such as C and Java and many other imperative languages.

“They are imperative in the sense that they consist of a sequence of commands, which are executed strictly one after the other,” they explained before pointing out how functional languages differ.

“A functional program is a single expression, which is executed by evaluating the expression. Anyone who has used a spreadsheet has experience of functional programming. In a spreadsheet, one specifies the value of each cell in terms of the values of other cells. The focus is on what is to be computed, not how it should be computed.”

To make the point, they present a sort program which runs to six lines in Haskell and the C equivalent which occupies 24 lines, both being suitable spaced for clarity.

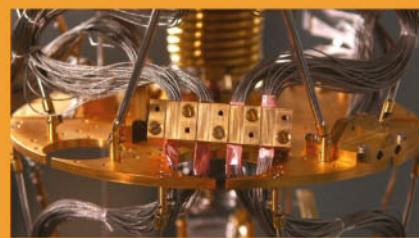
As with many programming languages, you can try Haskell out in your browser, specifically at <http://tryhaskell.org> and you can access a quick tutorial there. There are

QUANTUM PROGRAMMING

D-Wave machines

D-Wave, the manufacturer of the world’s only commercial quantum computer, offers several ways of programming its hardware. These range from common high-level languages, through languages designed to solve optimisation problems – the main strength of D-Wave machines, to the low-level Quantum Machine Language, the equivalent of the machine code used to program early computers.

Employing a so-called quantum annealing architecture, D-Wave machines can be thought of as analogue quantum computers. They are being used by Google and NASA to solve certain types of problems but they are not universal as our digital computers are. Meanwhile,



Above New programming languages are being developed for quantum computers

researchers are intent on bringing us universal, digital quantum computers, with the phenomenal power they will provide, and languages to support this hardware are already being developed.

For more than a decade, a team of researchers at Microsoft has been working on universal quantum computing, a model of computation that employs qubits instead of bits. Qubits can store binary 0s and 1s simultaneously, which could potentially lead to hugely parallel processing. Needless to say, using this bizarre sounding architecture requires a very different approach to programming.

Just a couple of months ago, Microsoft announced a new language designed for developers to create apps to debug on quantum simulators today and run on an actual topological quantum computer in the future. Fortunately, they say that you don’t have to be a quantum physicist to use the new technology.

According to a company spokesperson, “The new programming language is deeply integrated into Visual Studio, and it includes the kinds of tools that developers rely on for classical computing, such as debugging and auto complete.”

also several compilers available for various operating systems including Linux.

Consider also...

A list of the other new languages that are attracting attention would be a long list indeed. However, if you want to continue this tour of discovery, the following languages are well respected: Kotlin, Scala, TypeScript, Clojure, Rust, Swift and Julia.

Drawing up an ordered shortlist would be tricky but if you want to try your hand at a language that’s both up-and-coming and unusual, in the sense of employing a

somewhat different programming paradigm, how about trying Clojure? As a dialect of the old Lisp language, introduced back in 1958, it is a so-called list processing language.

Quoting Rich Hickey, Clojure’s author, “Clojure is a dynamic, general-purpose programming language, combining the approachability and interactivity development of a scripting language with an efficient and robust infrastructure for multithreaded programming. Clojure is a dialect of Lisp, and shares with Lisp the code-as-data philosophy and a powerful macro system.” ▶

1991

1995

2009

2017

Building on BASIC, Visual Basic is designed for ease of use, tailored for developing programs for GUIs

PHP and JavaScript appear. Improving facilities for server- and client-side programming, they bring more interactivity

Go is introduced with support for concurrent programming. It’s one of today’s top up-and-coming languages

Microsoft releases a language for quantum computers, maybe the most significant progress in programming for 60 years

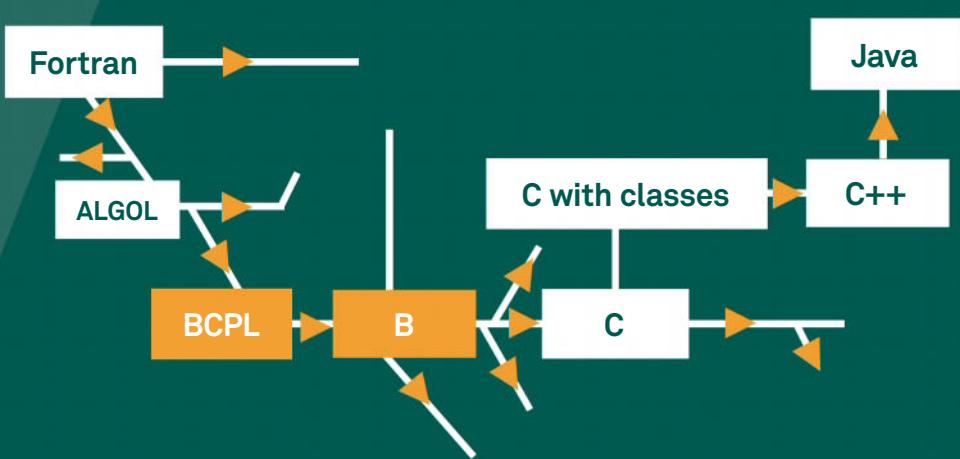
Future of programming

What does the future hold for programming languages?

Before looking at the future of programming languages, it would be helpful to delve into the past so we can understand the context. The very first computers were programmed in machine code. Initially, the program was entered by writing binary numbers to specific addresses using switches on the computer's front panel, although punched paper tape was soon used instead.

For example, to jump to address 0011100 on an old 8-bit processor, you might have to enter the op-code (the binary number that represents the instruction) 00001110, to one particular address and 0011100 into the next address.

Needless to say, this wasn't at all intuitive by today's standards and was very error-prone. First of all you needed to look up the op-code correctly, and then you'd need to accurately work out the address to jump to. What's more, maintaining or modifying code wasn't at all simple. For instance, if you added an instruction somewhere in the code, all subsequent instructions would have to be written to different addresses and any jump or branch instructions that



Above Today's programming languages, such as C++, are the result of evolutionary developments over 60 years.

might change from 00001110 0011100 to JPM INITIALISE.

Not only was this much easier and more intuitive to write and read, but code maintenance was less error-prone since the addresses referred to by labels would change automatically as instructions were added or removed and, similarly, the addresses referred to by variable names would also change as variables were added or removed.

Even so, an inherent drawback with machine code and assembly language is that the

set of instructions available is unique to a particular computer architecture. This means that programs written for one machine couldn't run on another.

“A single high-level instruction would often replace several machine code or assembler instructions with clear benefits **”**

referred to those instructions would be to be modified to refer to their new addresses.

Next came assembly languages, which were designed to overcome the drawbacks of machine code. This was achieved using software called an assembler that would translate assembly language code into machine code. This provided several advantages. First, instead of using an op-code, a meaningful mnemonic was used. In addition, variable names were used to refer to memory locations and instruction could be identified by labels. That first generation jump instruction, therefore,

QUICK TIP

The patient programmer

Early programmers were patient. They wrote their programs on a stack of punch cards and gave them to an operator. A couple of hours later, they'd be told there was a 'syntax error'.



Above We've come a long way from entering machine code via a computer's front panel switches

Furthermore, learning to programme on one computer didn't stand you in good stead for programming on another.

The next major development was of high-level languages, which were designed to overcome the issue of non-portability that applied to machine code and assembly languages. These languages have also been referred to as third-generation languages (and, retrospectively, machine code and assembly languages became first- and second-generation languages), although the generation terminology isn't recognised by all programming language researchers.

Instead of using the built-in instructions of a particular machine, high-level languages provide a general set of instructions that would be translated by a compiler into the instructions supported by the target machine. Other benefits were also on offer as the term 'high-level'

suggests. In particular, a single high-level instruction would often replace several machine code or assembly instructions with clear benefits to the programmer.

So, for example, to add the contents of two memory locations with the variable names A and B, storing the result in memory location B, might require the following three assembly instructions: LOAD A, ADD B, STORE B. In a typical high-level language, these three instructions would reduce to a single instruction such as B = A + B.

The concept of fourth- and fifth-generation languages goes back quite some time but these tend to be somewhat specialised and the vast majority of the languages now in use for general-purpose programming are third generation. Because the phrase isn't universally accepted and, according to one expert, is mostly used as a marketing gimmick by suppliers of some

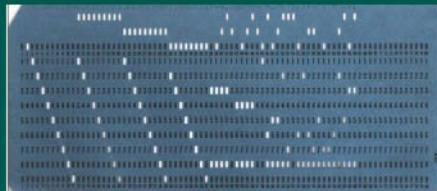
“Testing is a good way of checking out a program but it's impossible to find every bug”

languages, we won't look in detail at how fourth- and fifth-generation languages are defined. However, if we do think in terms of generations, the general consensus is that each generation aims to provide a means of defining a problem that is more removed than the previous generation from the workings of the hardware as we'll see later.

Evolutionary development

Machine code and assembly language are different because they were tied to particular computer architecture. However, with the introduction of high-level languages we can trace evolutionary paths that link today's commonly used languages with some of the very first ones, dating back to the late 1950s. As an example, we'll trace the ancestry of Java, today's most used language. The lineage is based on a study undertaken by IBM Research, the Retrocomputing Museum, Microsoft and Stanford University.

Fortran, the first high-level language to achieve popular support, was designed for scientific programming. Flow control was pretty much limited to GOTO, DO loops, functions and sub-routine calls and this led to 'spaghetti code'. ALGOL addressed this issue by providing support for structured



Above The punch card epitomises the early days of programming in languages like Fortran and COBOL

programming by allowing code to be arranged in blocks by use of the BEGIN and END statements. This soon gained widespread support and led to several spin-off languages, most importantly, from our perspective, a now largely forgotten language called BPCL which led to B which, in turn, inspired C.

Although undoubtedly a high-level language, C added low-level features such as bit manipulation functions. This made it suitable for efficient system programming, but it gained widespread support far beyond this initial niche. C++ built on C by adding object orientation and

this, in turn, was a direct predecessor to Java, which has a similar syntax to C++ but introduced the concept of the virtual machine for the ultimate in portability.

The way ahead

To get a view for what the future may hold, we spoke to two experts in programming language research – Professor Colin Runciman of the University of York and Dr Martin Lester of the University of Oxford.

A key issue raised by both is those developments aimed at making code more reliable. Testing is a good way of checking out a program but it's impossible to find every bug this way and errors often only come to light when the software has been distributed. Static analysis – detecting errors at compile time – is also limited.

Type systems are a classic way of discovering errors at compile time and a lot of current research is being conducted in this area with the aim not just of finding obvious errors like adding a text string to an integer, but less obvious ones such as writing a loop that will never terminate. One product of this line of research has been optional static type systems for languages that are traditionally dynamically typed. Examples include TypeScript for JavaScript and mypy for Python. ▶

Q&A

Interview: Martin Lester, University of Oxford

Martin Lester is a researcher at the University of Oxford with an interest in the theory and application of programming languages.

Today's top languages are mostly general purpose, imperative and object-oriented, so does that paradigm represent the pinnacle of development?

Just because a language is popular and widely used, it doesn't necessarily mean that it's been chosen for a good reason, or indeed chosen at all.

If a developer contributes to an existing project, the language is already fixed. Most professional developers work on existing projects, so get no choice in what language they use.

So are you saying there is more to come? C++, Java and Python have all undergone a lot of changes in the last ten years or so. Whenever a new feature is added to a language, the benefit is usually one of being able to write programs more quickly, have them run more quickly, or reduce the number of errors in them. So although I don't know what new features or paradigms are going to arise, I'm fairly confident that those will be the benefits.

What progress is being made in expressing problems in natural languages like English?

The problem is that our use of language is inherently ambiguous and contextual. If you try to have a discussion with even the best online chatbot, you'll probably find them pretty lacking – although they usually manage, or at least try, to cover this up by being deliberately vague and noncommittal.

If we ever manage to solve the hard AI problem of interpreting natural language, it won't matter how good our program synthesis is or which programming language we use, as the same AI will be able to write programs in whatever language we ask them to. However, until that time, we're definitely still going to need human programmers.



Above The success of Amazon's Alexa tech might suggest that engaging with computers using natural language may replace programming. But making queries is one thing; solving computational problems is quite another

A second area is support for the increasingly parallel nature of computers. "On the one hand, you have the scientific computing community, who are looking at how best to shift their computation onto massively parallel GPUs using CUDA or OpenCL," says Lester.

"On the other hand, people try to write their server applications to scale well to multi-core, multi-processor systems, whether in their own data centres or in the cloud. Two languages I hear people talking about a lot here are Go and Rust, which have different approaches to memory-safe concurrency. Go supports this based on having many processes that share data by passing messages, rather than by sharing

programming, namely constraint: "Most of the research is focused on how to find solutions rather than on developing the language in which to describe the solutions. There is a wide range of constraint solvers and a correspondingly large range of input languages. Unfortunately, some of the solvers are really good at solving certain kinds of problems but hopeless at others. So if you want to use constraint programming for a particular problem, you might have to try several solvers to find one that works well."

"The SMT-LIB project has produced a standard input language for constraint solvers, which all the major solvers support, and a library of benchmarks, so you can try to compare them and find out which works best for your problem. But the input language is quite low-level, so what often happens is that people write a program to generate the input. In terms of theoretical

developments, researchers will doubtless make their solvers better at solving harder problems more reliably, but I will be more interested to see to what degree constraint solvers become used as part of mainstream applications software."

And, finally, if you're worried that current research might put you out of work as a programmer, fear not – neither of the experts believe that programmers will be on the scrap heap anytime soon. ■

“ Neither of the experts we interviewed believe that programmers will be on the scrap heap anytime soon. ”

variables and using locks. Rust, meanwhile, has a clever compiler that checks shared variables are used safely." Needless to say, this is a major area of ongoing research.

More fundamentally, though, another strand of research mentioned was declarative programming, where you state the problem as opposed to defining the sequence of instructions needed to solve it.

Lester explained the challenges of research into one class of declarative

Q&A

Interview: Professor Colin Runciman, University of York

Colin Runciman works at the Department of Computer Science at University of York with research interests in programming languages and systems, functional programming and software tools.

Given that many of today's languages date back many years and their roots go back 50 or 60 years, have high-level languages reached maturity or are there further developments to come?

Further developments should be expected in at least two areas. First, stronger type systems will increase static guarantees that certain kinds of failure cannot occur when the program runs, and that many desired correctness properties are verified. Second, more powerful methods will be developed to compile programs effectively for execution on multi- and many-core machines, including strategies for portable optimisation.

The terms fourth and fifth generation date back quite some time but such languages haven't made a huge impact. Is this going to change?

The impact of declarative programming, mainly functional and logic programming, is set to increase in my view.

Since some languages don't involve creating something that could be called an algorithm, are the days numbered for programmers? Will queries eventually be made just by expressing questions in natural language?

Query languages are one thing – languages for solving computational problems are quite another. Many queries can already be cast quite effectively in natural language, even if it is just a Google search. But natural language would be hopeless for programming something like a network communication protocol, a web browser, an optimising code generator, or a task scheduler in an operating system. We need programmers who are highly trained in the fundamentals of software composition, including algorithms, data structures, components and the principles of software correctness and portability.

US subscription offer

Get 6 issues FREE

When you subscribe*

FREE DVD TRY 2 UBUNTU SPINS
www.linuxuser.co.uk

LinuxUser & Developer

THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION

ULTIMATE RESCUE & REPAIR KIT

- Digital forensics • Data recovery • File system repair • Partitioning & cloning • Security analysis

2 TOP DISTROS TO TRY NOW!

ANTERGOS
Modern, elegant and powered by Arch

XUBUNTU
Based on Ubuntu and built for speed

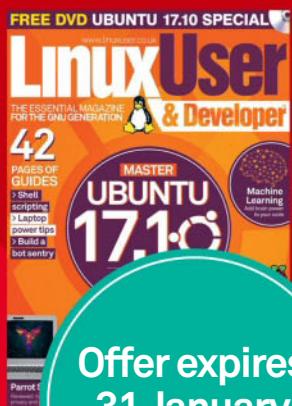
+ RESCUE & REPAIR TOOLS

SYSTEMRESCUECD AND RESCATUX
Salvage your system with two great distros

into network data with **WIRESHARK**

FREE resource downloads in every issue

“ The open source authority for enthusiasts and developers ”



Offer expires
31 January
2018

Order hotline **+44 344 848 2852**

Online at **www.myfavouritemagazines.co.uk/sublud**

*This is a US subscription offer. 6 free issues refers to the USA newsstand price of \$16.99 for 13 issues being \$220.87, compared with \$112.23 for a subscription. You will receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: bit.ly/magtdc. Offer ends 31 January 2018.

TIME TO STEP OFF THAT TREADMILL

With so many demands from work, home and family, there never seem to be enough hours in the day for you. Why not press pause once in a while, curl up with your favourite magazine and put a little oasis of 'you' in your day.

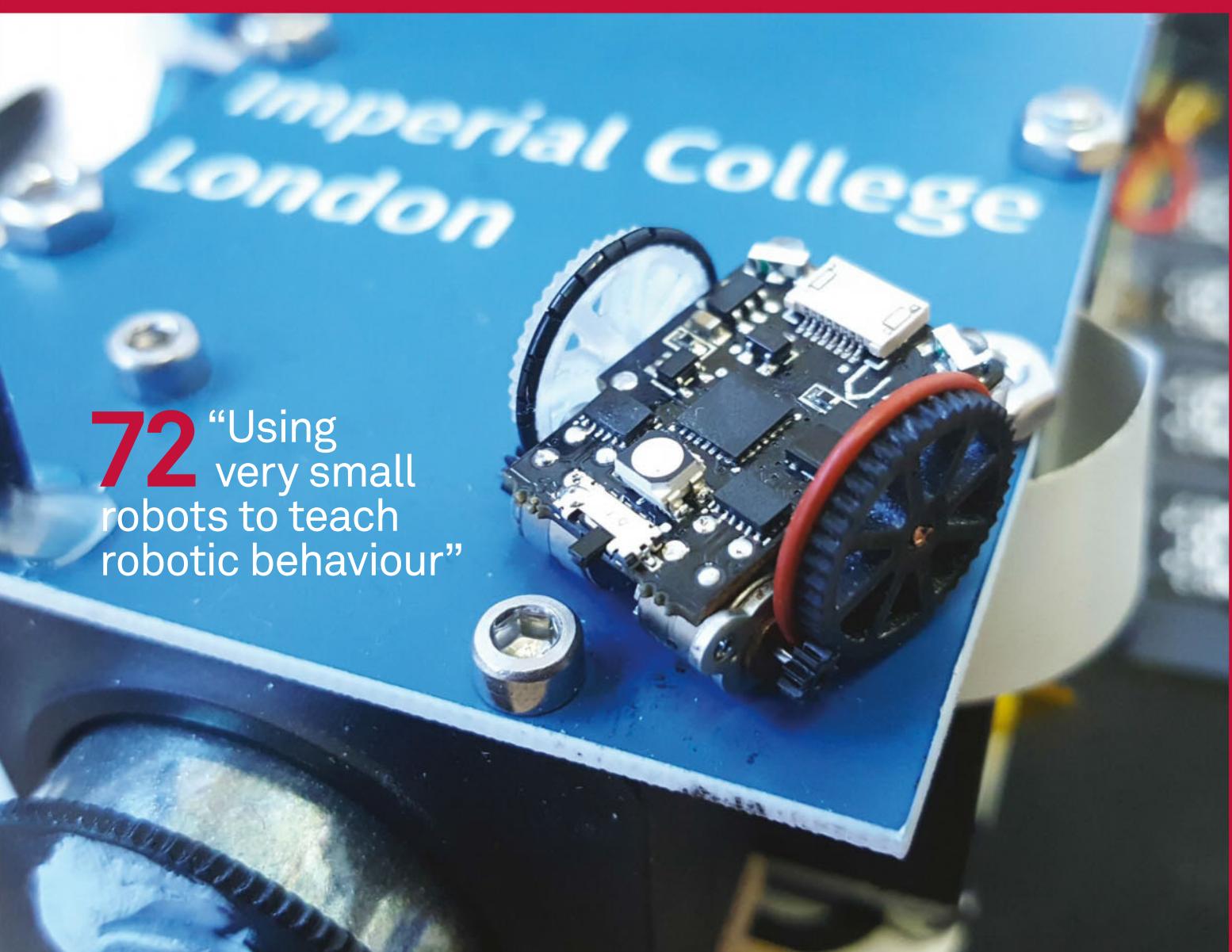


PRESS PAUSE
ENJOY A MAGAZINE MOMENT

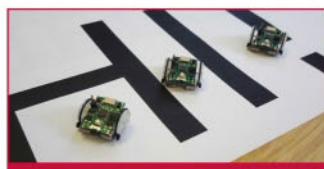
To find out more about Press Pause, visit:
pauseyourday.co.uk

PRACTICAL Raspberry Pi

72 “Using very small robots to teach robotic behaviour”



Contents



72 Pi Project: the micro robots are truly tiny



74 Draw vector shapes in your Minecraft world



76 Add voice-activated AI with Google Assistant



78 Create a lightweight database using SQLite



Joshua Elsdon
is currently a PhD candidate at Imperial College London, studying in the field of handheld robotics and augmented reality. He's been a keen tinkerer all his life, starting with audio engineering and high-voltage circuits as a teenager which has developed into a passion for robotics.

Like it?

Head to Joshua's YouTube channel for demos of his camera-based location system and very tiny robots following lines, performing synchronised patterns and generally whizzing about: <http://bit.ly/JEmicrorobots>

Further reading

To keep up to date with Joshua Elsdon's micro robots project and the release of version 6 of the kit for under £100, head to <http://bit.ly/HdayMicroRobots>

Micro robots

Micro robots

A pragmatic approach to teaching robotics has led to a project to build robots smaller than pocket change

Can you give us an overview of the micro robots projects? What's the idea behind your micro robots?

The micro robots project was formed when discussing how the Imperial Robotics Society could develop a course for teaching higher-level multi-robot behaviour. We have a very successful course introducing the basics of robotics on a small robot platform, roughly the size of an A5 sheet of paper, but robots of this size quickly become a problem if you want to control a load of them at once. The area you have to cordon off becomes prohibitive; also, generally you can only have one set that the class must share.

We decided that this course would not need access to the low-level hardware, as that would have been covered in the previous course, so we can use the full power of miniaturisation to reduce cost and size. We hope that in using very small robots to teach robotic behaviour

“ We use the robots for fun programming exercises ”

classes, we can have multiple systems available for testing and have to use less space for the arenas. Additionally, the low cost of highly integrated electronics that could be assembled automatically could lower the burden on volunteer instructors. Naturally, this seed for the project has given rise to a multi-year development effort for me and my hobby partner Dr Thomas Branch.

You've recently mentioned using a camera, QR and OpenCV for tracking the robots – can you explain how this works?

For most robotic experiments, knowing the location of individual robots is a fundamental piece of information. The issue is that the sensors on board the robots are not sophisticated enough to discover their location from the environment. So a typical solution is to have an overhead camera keep track of where the robots are to provide input to the navigation algorithms. With a fixed camera this can be achieved reasonably simply as the system's coordinates can be based relative to the camera and the size the robots will appear in the image can be hard-coded. Though due to the fun I have had whipping the robots out at opportune moments, I wanted the system to be possible to deploy completely ad hoc. Therefore, we have implemented a system that uses a QR code like marker in the scene as a coordinate reference that provides the computer vision system with a sense of scale and orientation. The camera does not need to be orthogonal to the surface, and we can even use the location of the camera as input to the system.

You also mention using the Raspberry Pi 3; how does that fit into project?

Originally we were thinking of this as a business case for providing educational kits, which are very price sensitive. Using the Raspberry Pi jumped out as a method of supplying the complete computational system with no setup for the user. We were aiming for the cost price of a robotic system with six robots and master computer to be roughly £100. Though because we are still doing lots of development on the project, we primarily use a full desktop system, for convenience.

Have any interesting projects have come out of the micro robots project and the training you've been running at Imperial?

Currently the robots are not used in teaching at Imperial, though in the future we hope to change that. I am using them in my private tutoring sessions with two 13-year-old boys. We use the robots for fun programming exercises, and we use a larger version of the robots for teaching SMD soldering techniques. The primary guiding project is to implement robotic football, though I always try and let the students have input on where the project goes, so we will have to wait and see what we actually implement.

Can you tell us about the robot HAT you're working on?

We had a couple of spare pins on the programming connector for the robot, so we decided to break out an I^C interface for expansion boards. As a proof of concept, we are partially through implementing a TOF (time of flight) laser scanner on the front of the robot. Due to the precise nature of the stepper motors we use for the drive motors, we should be able to collect a 360-degree laser scan of the environment from a robot. This can be used for SLAM (simultaneous location and mapping) which, if we can pull it off, would be by far the smallest robots to complete this task.

You mention that v6 is ready for manufacture? Is there a kit coming out soon?

Yes V6.0 is more or less ready to go; it implements an accelerometer for our new idea, running the robots on walls. We have demonstrated the fact that the robots can drive on a ferromagnetic surface mounted to a wall; the accelerometer will provide all robots with a reliable absolute orientation relative to gravity. As far as kits, it seems unlikely that there would be a kit any time soon – everything you need to know is open source; only the batteries are a pain to get. We are likely to make a larger batch this year for a demonstration of the system, and perhaps that would lead to some opportunity where the robots can be supplied publicly.

Components

Robot

- STM32F031
- 2x forward-facing IR proximity sensors and 1x downward-facing IR line sensor
- 2x micro stepper motors
- IR uplink and downlink modulated at 38kHz

System

- ST-Nucleo based IR bridge for communication between master and robots
- Master Linux system (RPi or laptop)
- User input such as joystick

Custom wheels

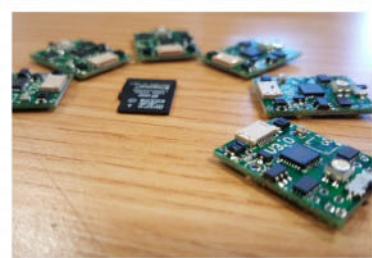
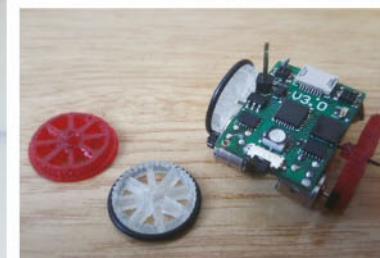
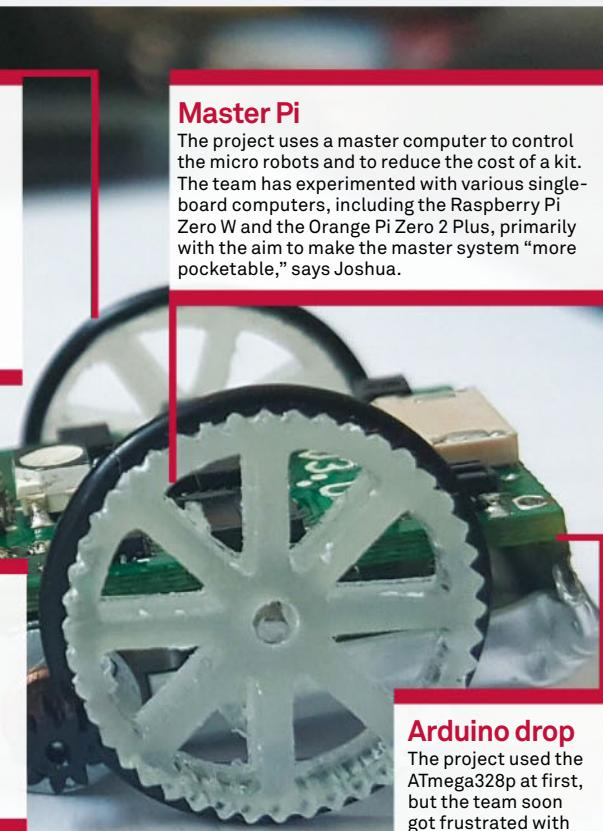
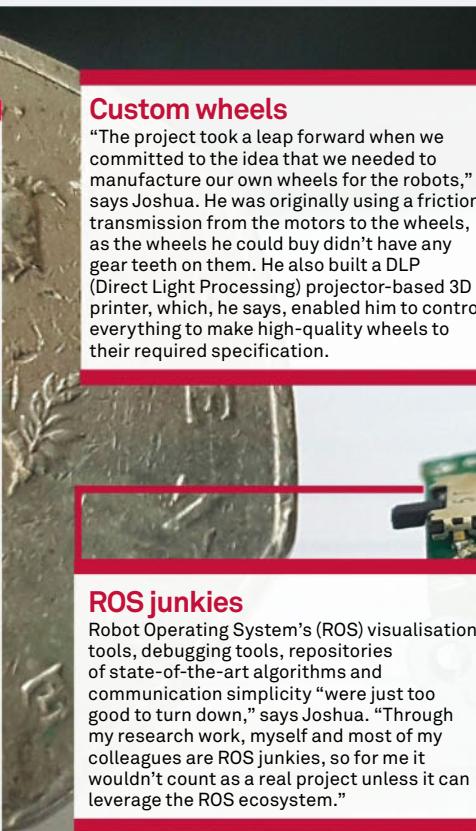
"The project took a leap forward when we committed to the idea that we needed to manufacture our own wheels for the robots," says Joshua. He was originally using a friction transmission from the motors to the wheels, as the wheels he could buy didn't have any gear teeth on them. He also built a DLP (Direct Light Processing) projector-based 3D printer, which, he says, enabled him to control everything to make high-quality wheels to their required specification.

Master Pi

The project uses a master computer to control the micro robots and to reduce the cost of a kit. The team has experimented with various single-board computers, including the Raspberry Pi Zero W and the Orange Pi Zero 2 Plus, primarily with the aim to make the master system "more pocketable," says Joshua.

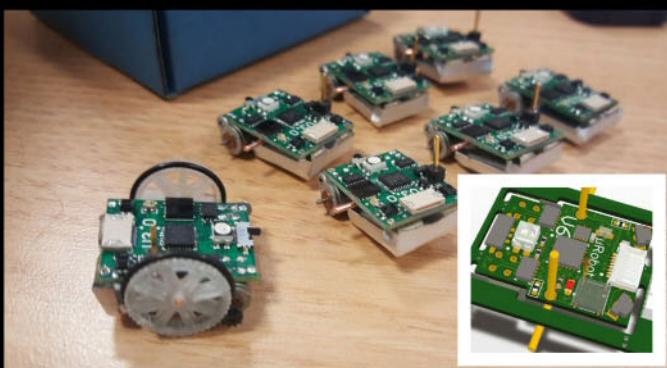
ROS junkies

Robot Operating System's (ROS) visualisation tools, debugging tools, repositories of state-of-the-art algorithms and communication simplicity "were just too good to turn down," says Joshua. "Through my research work, myself and most of my colleagues are ROS junkies, so for me it wouldn't count as a real project unless it can leverage the ROS ecosystem."



Arduino drop

The project used the ATmega328p at first, but the team soon got frustrated with the lack of resources in the Arduino IDE and so switched to the STM32L031. This offered plenty of timers to implement better motor control and have more flash and RAM. "[As] these robots are meant to abstract away from the low-level details for the user, using Arduino was probably misguided in the first place," says Joshua.



Above Version 6.0 of the micro robot kit is almost ready to go (rendered, above) and includes an accelerator for running the robots on walls: "We have demonstrated the fact that the robots can drive on a ferromagnetic surface mounted to a wall," Joshua tells us. "The accelerometer will provide all robots with a reliable absolute orientation relative to gravity."



Above A recent trip to a large robotics conference saw the micro robots well received, but did highlight a few problems: "The lesson learned [...] was that any demo of the project should be portable and quick to set up," says Joshua. For future trips he intends to integrate his calibrated system with controlled lighting and a mount for the camera into a single-board computer



Calvin Robinson

is Head of Computer Science at an all-through CofE state school in Barnet. Calvin also consults with schools all over London, providing high-quality Computing curricula.

Resources

■ McPiFoMo

<http://rogerthat.co.uk/McPiFoMo.rar>

■ Block IDs

<http://bit.ly/MC-BlockIDs>

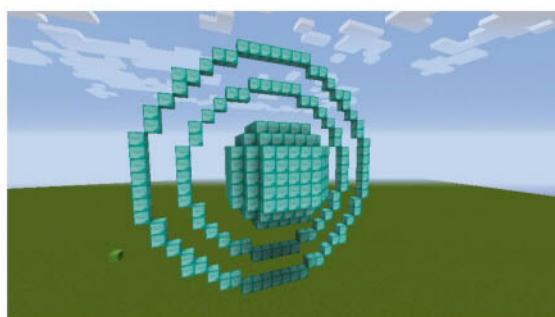
■ Minecraft Stuff

<http://bit.ly/MC-stuff>

Minecraft

Drawing vector shapes in Minecraft with Python code

Let's build some shapes in Minecraft using nothing but a few lines of Python code and the handy McPiFoMo hook



This issue we're looping back to where we started with the Python coding in Minecraft series in LU&D178. We'll be using Python code to draw shapes directly in our Minecraft worlds.

This tutorial is written with Minecraft Pi Edition in mind, but you don't have to be running Minecraft on a Raspberry Pi to follow along. We've put together a package that will work on any version of Minecraft, so you can run this tutorial on your favourite flavour of desktop Linux, Pi or no Pi. To allow Python to hook into Minecraft, you'll need to install McPiFoMo (see Resources) by extracting the contents of the `.minecraft` directory into `~/home/.minecraft`. McPiFoMo includes MCPiPy from MCPiPy.com and Raspberry Jam, developed by Alexander Pruss. Provided you have Python installed, no additional software is required, other than your favourite text editor or Python IDLE.

Martin O'Hanlon of stuffaboutcode.com has put together some prefabricated shape functions which we'll be using in this tutorial; they are available on GitHub (see 'Minecraft Stuff' link in Resources).

Python scripts in this tutorial should always be saved in `~/home/.minecraft/mcpipy/`, regardless of whether you're running Minecraft Pi Edition or Linux Minecraft. Be sure to run Minecraft with the 'Forge 1.8' profile included in McPiFoMo for your scripts to run correctly.

01 Prerequisites

Get yourself a copy of 'Minecraft Stuff':

```
git clone https://github.com/martinohanlon/
minecraft-stuff
```

Or just visit GitHub and manually download `minecraftstuff.py`, as that's all we'll need for this tutorial.

Just make sure you pop it in the `~/home/.minecraft` directory along with McPiFoMo.

Another way of installing 'Minecraft Stuff' would be to use Python's package index tool with `sudo pip install minecraftstuff` or `sudo pip3 install minecraftstuff`.

02 Python prep

We're going to want to create a new .py file in our favourite text editor / IDLE and import all the relevant Minecraft related Python modules.

```
import mcpi.minecraft as minecraft
import mcpi.block as block
import server
import time
import minecraftstuff
```

03 Connecting Python to Minecraft

We'll want to connect to our Minecraft world:

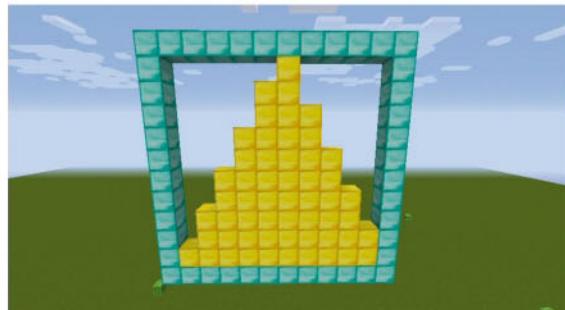
```
mc = minecraft.Minecraft.create(server.address)
```

Here we're creating an instance with the variable `mc` that we can use later on to spawn shapes directly into our open world. We'll want to use this variable when initiating an instance of 'Minecraft Stuff': `mcdrawing = minecraftstuff.MinecraftDrawing(mc)`.

Now let's track our current location in-game and we're good to go: `playerPos = mc.player.getTilePos()`.

04 Spawning shapes

The prefabricated shapes we have available to us are: `drawLine`, `drawSphere`, `drawCircle` and `drawFace`. We can spawn these shapes in our world with



the `mcdrawing` variable we set up a moment ago, i.e. `mcdrawing.drawSphere()`, but we'll need to make sure we pass the coordinates to the function, as well as the type of blocks we'd like to use. For example:

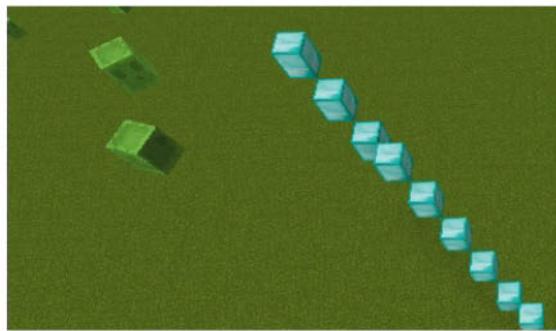
```
mcdrawing.drawSphere(playerPos.x, playerPos.y, playerPos.z, 15, block.DIAMOND_BLOCK.id)
```

05 Functions and their parameters

We have four different shape functions:

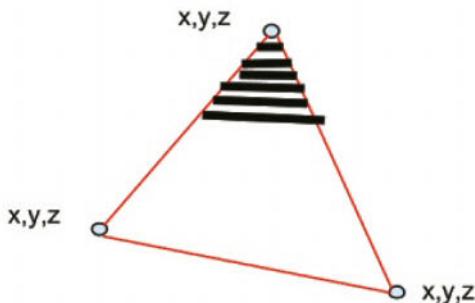
```
mcdrawing.drawLine(x1,y1,z1,x2,y2,z2,blockID)
mcdrawing.drawSphere(x,y,z, radius, blockID)
mcdrawing.drawCircle(x,y,z, radius, blockID)
mcdrawing.drawFace(shapePoints, True, blockID)
```

We'll expand on the `drawFace` function in a moment. It's a great tool for filling in surfaces. Think of `drawFace` as a paint bucket from Paint / Photoshop.



```
mcdrawing.drawLine(playerPos.x, playerPos.y + 2, playerPos.z, playerPos.x + 2, playerPos.y + 2, playerPos.z, block.DIAMOND_BLOCK.id)
```

Combined with `drawFace`, this provides some powerful tools to get creative in our Minecraft worlds, without spending too much time placing blocks.



06 Custom vectors with drawFace – part 1

The prefab shapes are great, but you might want to draw a custom shape of your own. We can do that by setting coordinates of each point and then filling in the blanks with `drawFace`.

```
shapePoints = []
shapePoints.append.minecraft	Vec3(x1,y1,z1))
shapePoints.append.minecraft	Vec3(x2,y2,z2))
shapePoints.append.minecraft	Vec3(x3,y3,z3))
```

07 Custom vectors with drawFace – part 2

Now that we've set the coordinates of the points, `drawFace` will connect up the dots and fill within the lines with your chosen `blockID`.

```
mcdrawing.drawFace(shapePoints, True, blockID)
```

For example: `mcdrawing.drawFace(shapePoints, True, block.DIAMOND_BLOCK.id)`.

However, if you merely want to draw the outline of the shape, you can change the Boolean from `True` to `False` and `drawFace` will create lines but not fill them in.

08 Crossing the line

Sometimes you might want to literally just create lines in any given direction, with a specific type of block. We can do that with the `drawLine` function:

09 A step in time

We imported the `time` module back in Step 2. That's so we can slow things down (or speed them up) when needed. When using the `drawLine` or `drawFace` functions, it's best to include a slight pause after each line, otherwise your game could have problems playing catch-up and the lag would result in a messy creation.

Just insert a `time.sleep(x)` after each instance, where `x` is an integer. A number between 1 and 5 should suffice.

10 Status update

As your Python script becomes more complex with the addition of different prefab and custom shapes, it's good practice to comment the code, but also a nice touch to update the current player with a status update.

```
mc.postToChat("Spawning X_Shape in the player world")
```

...where `X_Shape` is the name of what you're spawning. Insert these lines before (or after) creating each shape.

■ Python for Minecraft Pi

Using Python, we can hook directly into Minecraft Pi to perform complex calculations, alter the location of our player character and spawn blocks into the game world to create all kinds of creations, both 2D and 3D. We can program pretty much anything from pixel-art, to chat scripts that communicate directly with the player.

In this issue we create shapes, both prefabricated and completely custom vector graphics, that spawn in our world at the drop of a hat.

With each issue of **LU&D** we take a deeper look into coding Python for Minecraft Pi, with the aims of both improving our Python programming skills and gaining a better understanding of what goes on underneath the hood of everyone's favourite voxel-based videogame.



Nate Drake

is a technology journalist specialising in cybersecurity and Doomsday Devices.

Resources

■ ReSpeaker Pi drivers

[http://bit.ly/
GHReSpeaker](http://bit.ly/GHReSpeaker)

■ ReSpeaker 2-Mics pHAT

[http://bit.ly/
ReSpeakerpHAT](http://bit.ly/
ReSpeakerpHAT)

■ ReSpeaker documentation

[http://bit.ly/
ReSpeakerWiki](http://bit.ly/
ReSpeakerWiki)



Tutorial files available:
filesilo.co.uk

Google Assistant and ReSpeaker pHAT

Raspberry Pi AI

Use Google Assistant with your Raspberry Pi to endow it with voice-activated artificial intelligence



Google Assistant is a virtual personal assistant, designed to make your life easier by scheduling meetings, running searches and even displaying data.

In April this year Google released an SDK (software development kit), allowing developers to build their own Assistant-related hardware.

In this guide you'll learn how to integrate Google Assistant with your Pi and run a demo which can respond to your voice commands. As the Pi doesn't have a built in microphone, we've used the ReSpeaker 2-Mics Pi HAT, which is designed specifically for AI and voice applications. The ReSpeaker HAT costs just £12 and is compatible with the Pi Zero as well as the Pi 2B and 3B. However, if you have a USB microphone, you can use this instead.

By default, Google Assistant only responds to commands when prompted with a hotword – "Hey Google!". The privacy conscious will be pleased to hear that you can also program the ReSpeaker HAT to listen only when the on-board button is pressed. See the online documentation for tips on how to do this.

This tutorial assumes that you have a clean install of the most recent version of Raspbian (Stretch) on your Pi. Once the install is complete, make sure to open Terminal and run `sudo apt-get update` then `sudo apt-get upgrade` to bring your system fully up to date.

01 Connect the ReSpeaker

Mount the ReSpeaker 2-Mics Pi HAT on your Raspberry Pi, making sure the GPIO pins are properly aligned. If connected properly, the red LED will illuminate. Connect your headphones or speakers to the 3.5mm audio jack on the device. If your speakers aren't powered, connect another micro-USB cable to the power port on the ReSpeaker to supply enough current. The microphones themselves are located at either end of the HAT, labelled 'Left' and 'Right'.

02 Install device drivers

Return to the Terminal on your Pi and run `git clone --depth=1 https://github.com/reespeaker/seed-voicecard`. Switch to this directory with the command `cd seed-voicecard`, then launch the installer with `sudo ./install.sh 2mic`. Once installation is complete, reboot the Pi by running `sudo reboot`. Reopen a Terminal and run the command `aplay -l` to list all hardware playback devices. You should see the ReSpeaker listed under card 1 as 'seed2micvoicec'. Next, run the command `arecord -l` to list all sound capture devices to check that 'seed2micvoicec' is listed here too.

03 Configure sound device

Right-click the volume icon at the top right of your screen and make sure that the voicecard is selected. Right-click once again and choose 'USB Device Settings'. In the new window which opens, click the drop-down menu marked 'Sound Card' and choose the seed-voicecard or another preferred audio device. Next, click 'Make Default' to ensure the Pi will keep using this device next time it restarts. Don't worry for now about the lack of volume controls; simply click 'OK' to save and exit.

04 Adjust sound levels

Return to the Terminal and run the command `sudo alsamixer`. This handy utility lets you adjust the volume for playback and recording of all hardware devices. First press **F6** and use the arrow to select your chosen audio device, which should be listed under Card 1. Use **Return** to select it, then press **F5** to list all volume settings. Use the left/right arrow keys to select different controls and up/down to adjust individually. Press **Esc** to quit, then run `sudo alsactl store` to save the settings.



05 Visit Google Cloud Platform

In order to interface your device with Google Assistant, you'll need to have a Google account. If

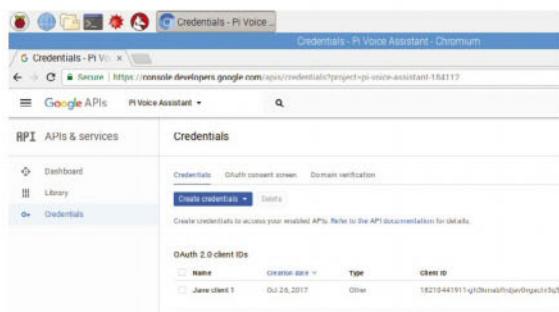
you don't have one, head over to <https://gmail.com> in your web browser of choice and click 'Create an Account' at the top right. Once you've done so, visit <https://console.cloud.google.com/project> and sign in with the account details. Click 'Yes', then 'Accept'. Next, visit <http://bit.ly/GoogleCP>. Click 'Enable'. A page will appear entitled 'To view this page, select a project.' Click 'Create' and name your project. Make a note of the project ID, then click 'Create' again.

06 Create product

In order to use the Google Assistant API, you need to generate credentials. These are stored in your Pi and let Google know you're using an authorised device. Click 'Enable' on the API page which has now loaded. Next, visit <http://bit.ly/GCP-OAuth>. Click 'Configure Consent' at the top right and enter a product name such as 'Jane's Pi Voice Assistant'. If you're creating a marketable application, fill in the homepage, logo and privacy policy fields. Click 'Save' at the bottom when you're done.

07 Create client ID

Google will return you to the 'Create Client ID' screen. Under 'Application Type', choose 'Other', then click 'Create'. You will now see your Client ID and secret key. Ignore these for now and click 'OK'. Google will now return you to the 'Credentials' screen where your client ID will now be listed. Click the down arrow on the right-hand side to save the credentials file with the extension .json. Once the file is downloaded, open your file browser and move it to `/home/pi`. This will make it easier to access.



08 Authorise Google Assistant

In your web browser, visit this address: <https://myaccount.google.com/activitycontrols>. Make sure to switch on 'Web & App Activity', 'Device Information' and 'Voice & Audio Activity'.

Next, open Terminal and run `sudo pip install --upgrade google-auth-oauthlib[tool]`. You can use this tool to authorise Google Assistant to work with your Pi, using the JSON file you downloaded in the previous step. To do this, run the command:

```
google-oauthlib-tool --client-secrets
/home/pi/client_secret_.json --scope https://
```

```
www.googleapis.com/auth/assistant-sdk-prototype
--save -headless.
```

Make sure to replace `client_secret_.json` with the actual name of the JSON file.

09 Enter confirmation code

The Google OAuth tool will now ask you to visit a URL to obtain a confirmation code. Right-click and copy the link. Open your web browser and right-click the address bar and choose 'Paste and Go'. Click on your own account name and then choose to 'Allow' the voice assistant. The page will generate a code. Copy this from your browser and paste it into the Terminal prompt. If successful, you'll see a message stating that your credentials have been saved.

10 Start Google Assistant demo

In Terminal, run the command `sudo apt-get install pulseaudio`, then launch it with `pulseaudio &`. Finally, run the command `google-assistant-demo`. Make

```
ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
  {"text": "what does the fox say"}
ON_RESPONDING_STARTED:
  {"is_error_response": False}
ON_RESPONDING_FINISHED
ON_CONVERSATION_TURN_FINISHED:
  {"with_follow_on_turn": False}

ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
  {"text": "what is 100 Euros in dollars"}
ON_RESPONDING_STARTED:
  {"is_error_response": False}
ON_RESPONDING_FINISHED
ON_CONVERSATION_TURN_FINISHED:
  {"with_follow_on_turn": False}
```

sure that your headphones or speakers are connected. Use the default hotwords "Hey Google" to have the Assistant listen, then ask it a standard question such as "What's the weather like in London today?" to hear a mini weather forecast.

Google Assistant can also play sounds; e.g. "What does a dog sound like?" It also performs conversions such as asking telling you the value of 100 euros in dollars. ■

Using virtual environments

If you want to use Google Assistant in an embedded device or application, you may wish to create a dedicated virtual environment to store the above demo separately to other projects.

Use Terminal to run `sudo apt-get install python3.4-venv`. Next, choose a name for your virtual environment and create it e.g. `python3 -m venv voice`.

If you decide to use a virtual environment, run the above commands after completing Step 7. You can switch to the virtual environment any time with the `activate` command e.g. `source voice/bin/activate`. Run `deactivate` to exit. To run a specific program you've installed in your virtual environment, enter the command as normal; e.g. `google-assistant-demo`.

Managing data in Python

When the amount of data you need to work with goes beyond easy flat files, it's time to move into using a database and a good place to start is SQLite



Joey Bernard
is a true renaissance man. He splits his time between building furniture, helping researchers with scientific computing problems and writing Android apps.

Why Python?

It's the official language of the Raspberry Pi. Read the docs at python.org/doc

In previous issues, we have looked at how to use data that is stored in files using regular file I/O. From here, we moved on to looking at how to use pandas to work with more structured data, especially in scientific work. But, what do you do when you have data that needs that go beyond these tools, especially in non-scientific domains? This is where you likely need to start looking at using a database to manage information in a better way. This month, we will start by looking at some of the lighter options available to work with simple databases.

"All of the SQLite code is part of your code"

In terms of lightweight databases, SQLite is the de facto standard in most environments. It comes as a C library that provides access to a file-backed database that is stored on the local disk. One huge advantage is that it does not need to run a server process to manage the database. All of the code is actually part of your code. The query language used is a variant of standard SQL. This means that you can start your project using SQLite, and then be able to move to a larger database system with minimal changes to your code.

There is a port to Python available in the module 'sqlite3' which supports all of the functionality. Because it is the standard for really lightweight database functionality, it is included as part of the standard Python library. So you should have it available wherever you have Python installed. The very first step is to create a connection object that starts up the SQLite infrastructure:

```
import sqlite3
```

```
my_conn = sqlite3.  
connection('example.db')
```

This gives you a connection object that enables interactions with the database stored in the **example.db** file in the current directory. If it doesn't already exist, the **sqlite3** module will create a new database file. If you only require a temporary database that needs to live for the duration of your program run, you can give the connection method the special filename ':memory:' to create a database stored solely in RAM.

Now that you have a database, what can you do with it? The first

step is to create a cursor object for the database to handle SQL statements being applied

to the database. You can do so with **my_cursor = my_conn.cursor()**.

The first database thing you will need to do is to create tables to store your data. As an example, the following code creates a small table to store names and phone numbers.

```
my_cursor.execute("CREATE  
TABLE phone  
(name text, phone_num  
text)")
```

You have to include the data type for each column of the table. SQLite natively supports SQL data types BLOB, TEXT, REAL, INTEGER and NULL. These map to the Python data types byte, str, float, int and None. The **execute** method runs any single SQL statement that you need to have run against the database. These statements are not committed to the file store for the database, however. In order to have the results actually written out, you need to run **my_conn.commit()**. Note that this method is part of the

connection object, not the cursor object. If you have a different thread also using the same SQLite database file, it won't see any changes until a commit is called. This means that you can use the **rollback()** method to undo any changes, back to the last time **commit()** was called. This allows you to have you a rudimentary form of transactions, similar to the functionality of larger relational databases.

Now that we have a table, we should start populating it with data. The simplest way to do this is to use a direct **INSERT** statement, e.g.:

```
my_cursor.execute("INSERT  
INTO phone VALUES ('Joey  
Bernard', '555-5555')")
```

While this is okay for hard-coded values, you'll probably have data coming from the user that needs to be entered into the database. In these cases, you should always check this input and sanitise it so there's no code that can be used for an SQL injection attack. You can do this, then do string manipulation to create the complete SQL statement before calling the **execute** method. The other option available is to use an SQL statement that contains placeholders that can be replaced with the values stored in variables. This makes the validation of the input data a bit easier to handle. The above example would then look like:

```
my_name = 'Joey Bernard'  
my_number = '555-5555'  
my_cursor.execute("INSERT  
INTO phone VALUES (?,?)",  
(my_name,my_number))
```

The values to be used in the SQL statement are provided within a tuple. If you have a larger amount of data that needs to be handled in one go, you can use the **executemany()**

What if SQLite isn't light enough?

function, available in the cursor object. In this case, the SQL statement is structured the same as above. The second parameter is any kind of iterable object that can be used to get a sequence of values. This means that you could write a generator function if your data can be processed that way. It is another tool available to automate your data management issues.

Now that we have some data in the database, how can we pull it back out and work with it? The basic SQL statement that is used is the **SELECT** statement. You can use the following statement to get my phone number.

```
my_cursor.execute("SELECT phone_num FROM phone WHERE name=:who", {"who":'Joey Bernard'})  
print(my_cursor.fetchone())
```

As you can see, you need to call some kind of fetching method in order to get your actual results back. The **fetchone()** method returns the next returned value from the list of returned values. When you reach the bottom of the list, it will return `None`. If you want to process returned values in blocks, you can use the cursor method **fetchmany(size)**, where size is how many items to return bundled within a list. When this method runs out of items to return, it sends back an empty list. If you want to get the full collection of all items that matched your **SELECT** statement, you can use the **fetchall()** method to get a list of the entire collection. You do need to remember that any of the methods that return multiple values still start wherever the cursor currently is, not from the beginning of the returned collection.

Sometimes, you may need to add some processing functionality to the database. In these cases, you can actually create a function that can be used from within other SQL statements. For example, you could

create a database function that returns the sine of some value.

```
import math  
my_conn.create_function("sin",  
1, math.sin)  
cursor2 = my_conn.cursor()  
cursor2.execute("SELECT sin(?)", (42,))  
print(cursor2.fetchone())
```

There is a special class of database functions, called aggregators, that you may wish to create, too. These take a series of values and apply an aggregation function, like summing, over all of them. You can use the **create_aggregate()** method to register a class to act as the new aggregator. The class needs to provide a **step()** method to do the aggregation calculation and a **finalize()** method that returns the final result.

One last item you may want to be able to do is to have larger blocks of SQL statements run against your data. In this case, you will want to use the cursor object's **executescript()** method. This method takes a string object that contains an entire script and runs it as a single block. One important difference here is that a **commit()** call is made just before your script is run. If you need to be able to do rollbacks, this is an important caveat to keep in mind. When you start to have more complicated queries, you may need to track where your results came from. The **description** property of the cursor object returns the column names for the last executed query.

When you are all done, you should always call the **close()** method of the connection object. But, be aware that a **commit** is not done automatically. You will need to call it yourself before closing. This ensures that all of your transactions are flushed out to disk and the database is in a correct state. Now you can add more robust data management to your code. ■

Sometimes, even SQLite may not be lightweight enough, depending on your specific requirements. In these cases, you do have another option. There is a very old file-backed database from the earliest days of UNIX, called `dbm`. `Dbm` databases store data as a set of key-value pairs within a file on the file system. To start, you will need to open a database with code like that given below.

```
import dbm  
db = dbm.open('example.db', 'c')
```

This opens the database in the file `example.db`, or creates it if it doesn't already exist. You can insert a value to a given key, or get a value based on a key. Below is an example of storing a name/phone number pair.

```
db['Joey Bernard'] = '555-5555'
```

When you do the query, you need to remember that everything is stored as byte strings, so you will need to use those to get values.

```
my_number = db.get(b'Joey Bernard')
```

There are two more advanced variants available, `gdbm` and `ndbm`. They each add some further functionality above and beyond that provided by the basic `dbm` implementation. One important thing to note is that the file formats for the different variants are not compatible. So if you create a database with `gdbm`, you will not be able to read it with `ndbm`. There is a function, named `whichdb()`, that will take a filename and try to figure out which type of `dbm` file it is. `Gdbm` has methods to easily allow you to traverse the entire database. You start by using the `firstkey()` method to get the first key in the database. You can then travel through the entire database by using the method `nextkey(key)`. `Gdbm` also provides a method, named `reorganize()`, which can be used to collapse a database file after a number of deletions.

Because `dbm`, and its variants, store data as sets of key/value pairs, it maps quite naturally to the concepts around dictionaries. You can use the same syntax, including the `'in'` keyword, from dictionaries when you work with any of the `dbm` variants. These modules allow you to have data stores that you can use to store simpler data structures within your own programs.



The source for tech buying advice

techradar.com

Reviews

81 Group test | 86 Hardware | 88 Distro | 90 Free software



GROUP TEST

Arch-based distributions

Experience the benefits of the venerable Arch Linux in the comforts of a desktop distribution that takes the pain out of that installation

Antergos

A multilingual distribution that's proud of the fact that it's designed with simplicity in mind. Antergos ships with custom artwork and claims it provides a ready-to-use system that doesn't require any additional steps once it's been installed.

<https://antergos.com>

Chakra GNU/Linux

One of the two most popular Arch-based distributions designed for desktop users, Chakra uses a half-rolling release cycle. The distro marks certain packages as core that only receive updates to fix security issues while other apps are on a rolling-release model.

<https://chakralinux.org>

Manjaro Linux

The other popular Arch-based desktop distribution, Manjaro, uses its own set of repositories. The distribution is available with three different desktops – Xfce, KDE and Gnome – and the latest release has been announced as the project's last to offer support for 32-bit hardware.

<https://manjaro.org>

RevengeOS

The oddly-named distribution is all about choice – starting from its six desktop environments to the default apps. RevengeOS' goal is to provide an easy-to-install Arch distribution while preserving the power and customisation offered by its Arch Linux base.

<http://revengeos.weebly.com>

Antergos

Makes good use of Arch's large repos to offer seven desktops



■ You can enable the Arch User Repository (AUR), in addition to Antergos' own repo, from the very useful Cnchi installer

Installation

The home-brewed, multilingual Cnchi installer does a good job of anchoring the distro. It allows you to choose from the seven supported desktops and can also install an Arch-like, CLI-only base. The partitioning step is intuitive enough for new users and also helps advanced ones use ZFS, setup LVM and home on a separate partition.

Pre-installed apps

Antergos has the usual slew of productivity apps. By default, the distro uses the Chromium web browser but you can install Firefox during installation where you also get options to pull in LibreOffice and Steam. The default Gnome desktop also includes Gnome tweaks and dconf editor to tweak the Gnome desktop.

Usability

Besides the project's Cnchi installer, there are no custom tools of the project. The distro allows you choose between cinnamon, deepin, gnome, KDE, Mate, Openbox and Xfce. All desktops come with their own configuration tools and they all have been tweaked to look very pleasing with a wide selection of desktop wallpapers, along with the Numix Frost theme.

Help & support

The project has very active multilingual forum boards with separate sections for installation, resolving newbie queries, applications and desktop environments, pacman & package upgrade issues, etc. There are several useful articles in the well categorised wiki and you can also seek advice on the project's IRC channel.

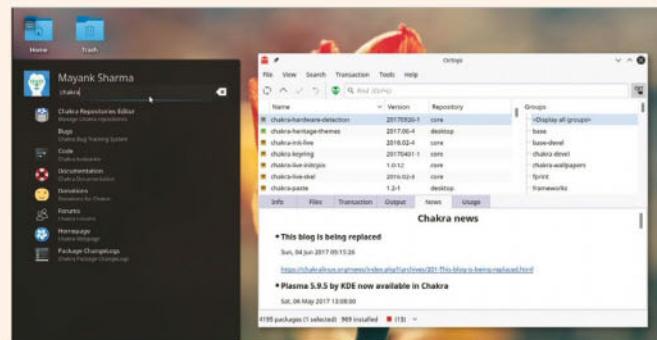
Overall

Antergos is an aesthetically pleasing distribution. Its Cnchi installer does a wonderful job and helps users customise various aspects of the installation including app selection.

8

Chakra Linux

A perfect rendition of KDE with an app selection to match



■ Chakra has several custom repositories, including a community-maintained repository that's inspired by the Arch User Repository

Installation

Chakra uses the distribution-independent Calamares installer. It's intuitive and can be navigated with ease. Like Cnchi, you can ask Calamares to automatically partition the disk or let you modify partitions manually. However, Calamares doesn't offer as many options as Cnchi installer.

Pre-installed apps

Chakra expressly bills itself as a KDE-centric distribution, which is why it includes a whole gamut of KDE apps including the complete Calligra office suite. It also includes other Qt apps such as the Bomi media player and the lightweight QupZilla browser in place of a more feature-rich browser like Firefox.

Usability

Chakra uses Octopi for package management and for keeping your installation updated. It also includes a cache cleaner and a repository editor. In terms of overall usability, KDE has always had a tendency to overwhelm the new user, which makes Chakra best suited for those who are already familiar and comfortable with the desktop.

Help & support

The project has recently overhauled its support infrastructure and the developers now engage with the users via the community section on the website. You can still find the link to the old wiki that has several useful articles to help install and manage the distribution, while others only have skeletal content.

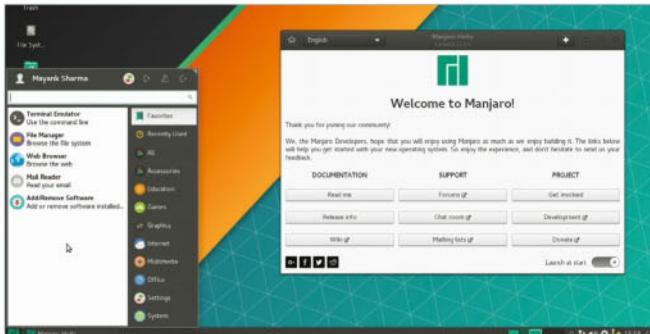
Overall

An Arch-based, 64-bit-only rolling release distro, Chakra's objective is to give its users the ultimate KDE experience. It's an excellent desktop distro for those whose demands align with its goals.

7

Manjaro Linux

Helps all users leverage the power of Arch with its powerful custom tools



With MSM you can install proprietary drivers for connected hardware and even switch between available kernels with a single click

Installation

Manjaro is available in three officially supported flavours and while the project doesn't recommend one over another, we've used the Xfce edition that is listed first on the download page. All editions use a customised Calamares installer that enables you to encrypt the partition in which you plan to install the distribution.

Pre-installed apps

Manjaro includes all the usual mainstream popular apps such as the LibreOffice suite, GIMP, VLC, Firefox, Thunderbird, Steam client, etc. The remaining space is stuffed with handy utilities from the Xfce stable. If you need more apps from the repos, the Xfce edition uses Pamac for package management and handling updates.

Usability

Manjaro is very usable since it bundles the regular apps instead of esoteric alternatives. The Xfce desktop follows the conventional desktop metaphor and will suit a large number of potential users. Another plus is the project's custom tools – especially the Manjaro Settings Manager – that allow users to take advantage of the Arch base.

Help & support

Manjaro trumps the lot with a channel on Vimeo with over two dozen videos on various aspects of the distribution and its development. Their wiki is fairly detailed and well categorised, and the project hosts multilingual forums and IRC channels. The distro also bundles a detailed PDF user guide.

Overall

Manjaro's developers have gone to great lengths to help users experience the power of Arch. With its custom tools and bundled apps, the ready-to-use desktop scores highly in both form and function.

9

RevengeOS

A lightweight distribution with tools to help you choose and customise



With the distro's Software Installation Tool you can install popular productivity and multimedia apps with a single click

Installation

RevengeOS uses its own home-brewed Nemesis installer. It offers three types of installs – Normal, OEM, and StationX. The installer is fairly intuitive to navigate and allows you to choose between six desktop environments. The partitioning step offers an automatic installation option but just fires up Gparted for manual partitioning.

Pre-installed apps

During installation you get options to install certain useful software such as LibreOffice, Wine, Steam client, etc. Besides these, the distro contains the usual collection of apps that ship with the desktop you've installed. We opted for its customised OBR Openbox desktop that installs several Xfce tools and some extras like the Gufw firewall.

Usability

The distro boots to a welcome screen that points to scripts to remove the pre-installed VirtualBox modules and another to install proprietary Nvidia drivers. There's also a custom control centre for accessing various custom tools for managing all aspects of the system, from changing wallpapers and docks to installing proprietary codecs and switching kernels.

Help & support

The welcome screen has links to join the project's Google+ community page and another to the distro's fairly active forum boards. Here you'll find boards for posting problems regarding installation and any issues with the apps. The website also has a wiki with some articles that you may find useful.

Overall

Overflowing with custom tools and scripts, RevengeOS is a well-crafted distribution that does a wonderful job of handholding new users and helping them get to grips with Arch.

8

In brief: compare and contrast our verdicts

	Antergos	Chakra GNU/Linux	Manjaro Linux	RevengeOS				
Installation	Its installer helps you pick a desktop and has smart partitioning options.	8	Uses the Calamares installer that's not as functional as Cnchi but works well.	7	The Calamares installer offers users the option of encrypting the installation partition.	8	Its Nemesis installer is very usable and supports three types of installs.	8
Pre-installed apps	You can use the installer to grab apps like Firefox, LibreOffice & Steam.	8	Takes pride in being a KDE-only distro and packs in all sorts of Qt apps.	8	Includes all of the popular apps you'd expect in a modern desktop distro.	9	The installer helps you to pick the desktop as well as popular, everyday apps.	8
Usability	All of the supported desktops have been tweaked to look good and are very usable.	8	The inclusion of Qt-only apps over some mainstream ones can be an issue for some.	7	The custom tools and the bouquet of apps make it easy to use and tweak.	9	You can mould the installation as per your needs with its custom control centre.	9
Help & Support	Offers enough options to help you find your way around any possible issues.	8	It's new but the recently overhauled support section will find you a solution.	7	The usual text-based options are complemented by videos on the Vimeo channel.	9	You get a Google+ page and a couple of boards on the forum to post your queries.	7
Overall	A smart-looking distro with an installer that helps customise the install.	8	The ultimate Arch-based distro for users looking for a pure KDE experience.	7	Allows users of all skill levels to experience the power and flexibility of Arch.	9	Essentially a one-man show that includes a great many tools to shape the installation.	8

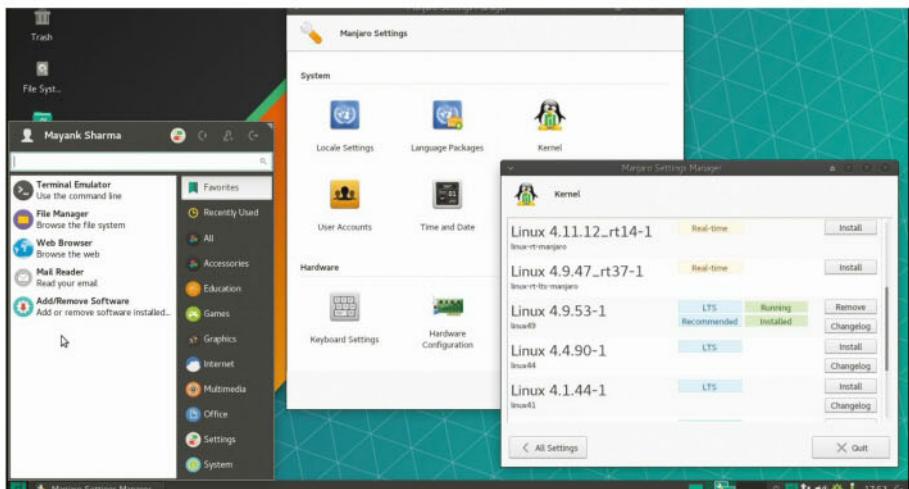
AND THE WINNER IS...

Manjaro Linux

After Ubuntu, Arch Linux is one of the most popular projects that's used as a base for creating all kinds of niche as well as general purpose distributions. In this group test, we've included the ones that focus on helping new and inexperienced users uncover the power of Arch. These distros help save them the time and effort investment that's typically required in deploying an Arch install.

Chakra ruled itself out of contention for its affection towards KDE users, which, with its myriad of options, isn't always the best environment for inexperienced users. RevengeOS and its gamut of custom tools make for a pleasant experience but the distro is a one-man show without a fixed release schedule. Plus if you don't like the default theme, tweaking the visuals will take some time and effort that's best spent elsewhere.

In the end, the real contest – and a close one at that – was between Antergos and Manjaro. A big plus for Antergos is that it offers the option of multiple desktops during installation. It's also visually appealing but apart from being a Cnchi installer, there are no other custom tools.



Use the CLI Manjaro architect tool to install a customised build with your choice of kernel, drivers etc

Manjaro, on the other hand, includes the very useful Manjaro Settings Manager that helps users take advantage of the Arch ecosystem without getting into the nitty-gritties. It's also chock-full of apps and can be used straight out of the box.

The project has its own dedicated software repositories that deliver thoroughly

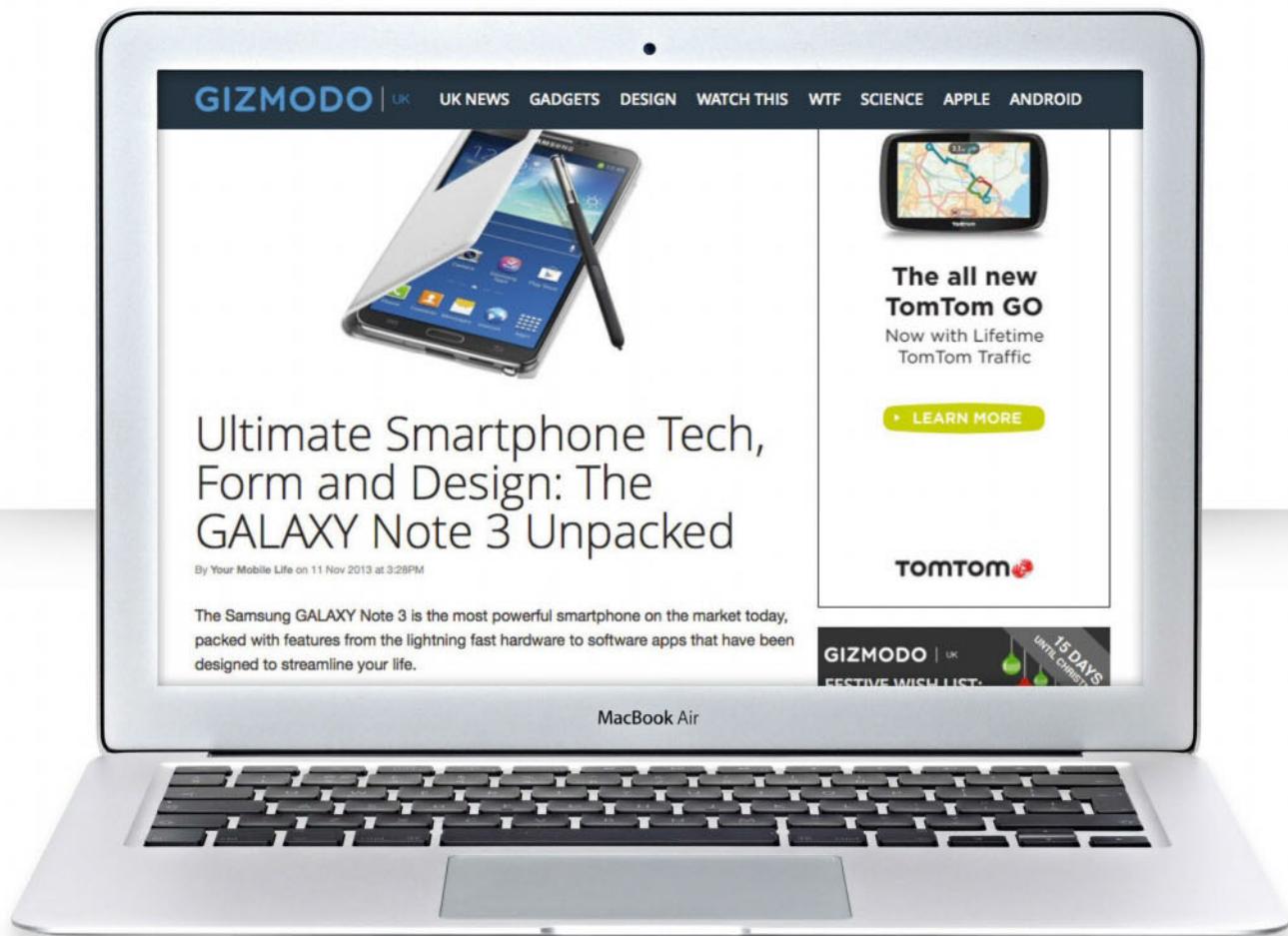
tested and stable software that works well in conjunction with Arch and AUR repositories and can be used for bleeding-edge software.

Finally, Manjaro remains one of the very few independently developed projects that is still on the market for both 32-bit and 64-bit architectures.

Mayank Sharma

GIZMODO | UK

Not your average technology website



EXPLORE NEW WORLDS OF TECHNOLOGY GADGETS, SCIENCE, DESIGN AND MORE

- Fascinating reports from the bleeding edge of tech
- Innovations, culture and geek culture explored
- Join the UK's leading online tech community

www.gizmodo.co.uk

 twitter.com/GizmodoUK

 facebook.com/GizmodoUK



HARDWARE

iStorage diskAshur Pro 2

Price

From £195 for 500GB (reviewed £489, 5TB)

Website

<https://istorage-uk.com>

Specs

Capacity: 500GB-5TB

Data transfer speed: Up to 148MBps (Read), 140MBps (Write)

Power supply: USB bus-powered

Interface: USB 3.1—up to 5Gbps

Hardware data encryption: Real-Time Military Grade AES-XTS 256-bit Full-Disk Hardware Encryption

Warranty: 2 years

Dimensions (W, D, H):

124mmx84mmx20

mm (500GB, 1/2TB),

124mmx84mmx28mm (3/4/5TB)

Weight: 500GB/1/2TB 225g, 3/4/5TB 331g

An expensive portable hard drive on the face of it, but good security has never been cheap

For those who carry sensitive information around with them on a daily basis, there's an ever-present concern with losing the device carrying that precious data (or worse, having it stolen).

This is concern addressed by the new iStorage diskAshur Pro2 external HD, a compact storage device designed to work with secure data without the need to install software on all the systems it will meet. The Pro2 retails at £489 (\$651), comes in 500GB, 1TB, 2TB, 3TB, 4TB and 5TB capacities, and it was the latter which the firm supplied to us for review. The Pro2 is expensive as 5TB external drives go, being quadruple what Seagate asks for its Backup Plus 5TB drive, and more than double

LaCie's rugged Thunderbolt 5TB models. But after removing the drive from the packaging, we realised why iStorage asks so much for it, because this is, without doubt, one of the most glorious pieces of product engineering we've had the pleasure to handle. The upper and lower surfaces are cool-to-the-touch metal, and the waistband is soft-textured rubber.

The drive is just 84mm wide, 124mm long and 20mm deep, dictating that this uses 2.5-inch drives internally to provide 5TB of capacity. As if to underline how much of the cost goes into the engineering and not the capacity, the 2TB model is £329 (\$437 approx), only £160 (\$212 approx) cheaper.



“The unit has numerous hardware safeguards to defend against external tampering, bypass attacks and fault injections”

From a design perspective, two features make this drive special: the first is the built-in USB cable. The cable is only 12cm long when unclipped, but that's enough to attach it to a laptop or desktop PC. The other standout feature is the built-in numeric pad. This is an integral feature of the security mechanism iStorage has implemented. In addition to the numbers, there are a few special keys for operating the unit when it's attached to a computer.

Above the numeric pad are three LEDs that confirm the locked condition, and also show drive activity. At the heart of this design is a secure microprocessor (Common Criteria EAL4+ ready) that handles the encryption of the device. This, in theory, means that if the bare drive is extracted from its case, an attacker is no closer to getting to the data.

A data thief will need the 7-15 digit numeric password created when the Pro2 was last configured. If you fail to enter the correct code enough times then this will result in the drive deleting the encryption key, rendering the contents beyond reach forever, unless you know people in the security services who can crack AES-XTS 256-bit.

In addition, the unit also has numerous hardware safeguards to defend against external tampering, bypass attacks and fault injections. Should it detect any attempt to get into the case or tinker with USB, it will trigger a deadlock frozen state, at which point further assault is pointless.

Devices with a numeric pad like this usually come with a PC application that you'll need to install, but the Pro2 is fully self-contained. That allows it to work as well with Linux as many other OSes it supports, such as Android, Chrome, Mac OS and Windows. You can format it to whatever filesystem you use—even one you've created yourself.

The unit comes with a default admin PIN number defined, and you can change that directly using the pad. But the most Bond-esque PIN code you can define is the one that initiates a 'self-destruct' sequence, which performs an internal crypto-wipe where all the PINs and data are erased, and the drive must be reformatted before it can be used again. For the majority of people, the Pro2 has enough in the way of protection.

With 145.5MBps reads and 144.8MBps writes, the spinning disk inside the Pro2 has some intent about it. While an SSD would be quicker (and iStorage provides models with those inside, too), those performance levels are good enough, and about as rapid as a PC with hard disk-based storage is likely to be. The unit is also IP56 certified, making it water and dust-resistant, though not waterproof. An extra touch in terms of the physical protection is that the keys on the pad are coated in epoxy. The coating makes it harder to work out what keys are being used on a regular basis.

Mark Pickavance

Pros

Beautiful construction and offers genuine data security. Being self-contained, it will work with any Linux distro

Cons

You get what you pay for, so it's expensive and also potentially a little daunting to set up

Summary

A genuinely secure storage device that's built to handle physical abuse and nefarious decryption – but it comes at a premium price. The overall combination of a well-considered security model which is also a superbly engineered device is an alluring one.

9



Above Pop is available only for 64-bit architecture with separate releases for Intel/AMD and Nvidia hardware

DISTRO

Pop!_OS 17.10



RAM
2GB
Storage
20GB
Specs

CPU: 64-bit Intel or AMD processor
Graphics: AMD or Nvidia hardware

Available from: <https://system76.com/pop>

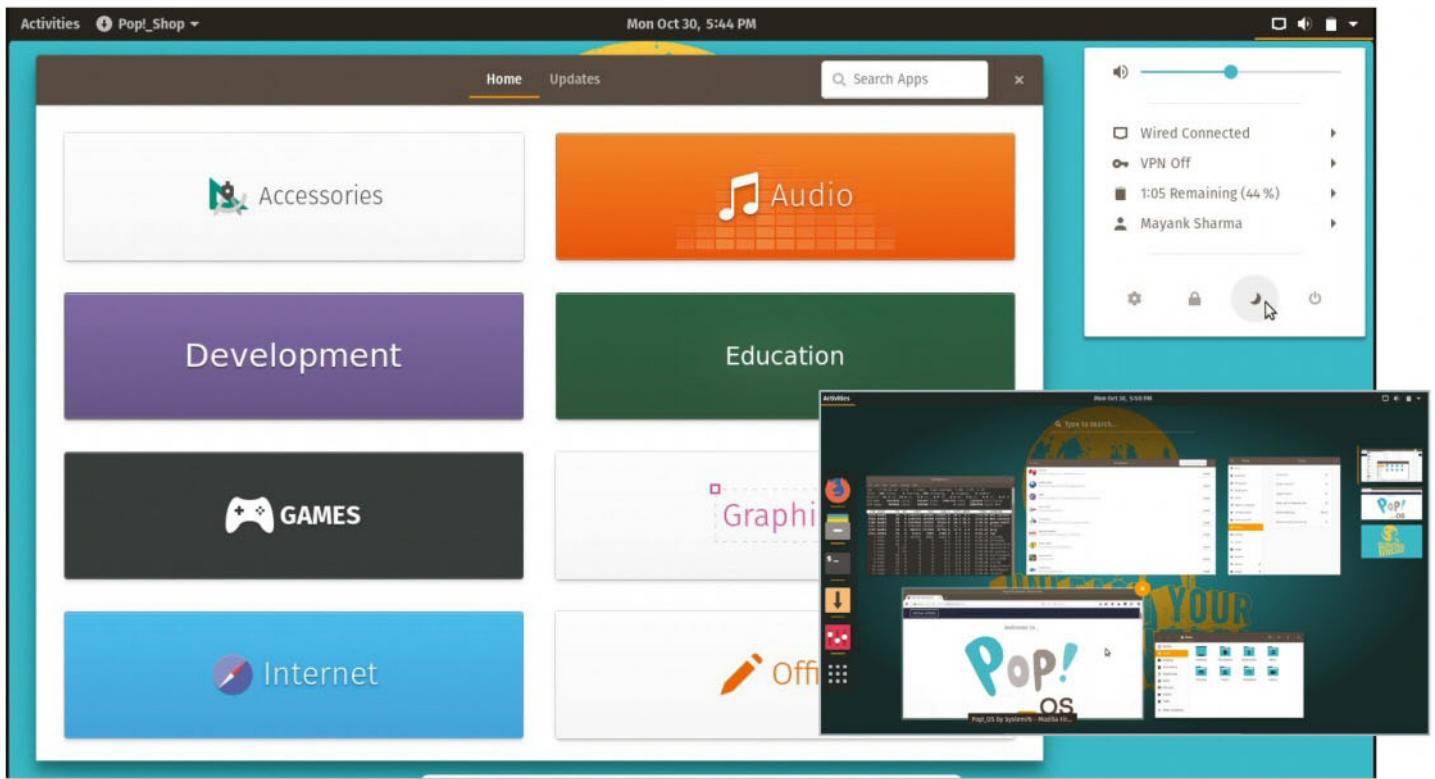
Caught out by the dramatic changes in Ubuntu, System76 decides to take matters into its own hands

System76 is one of the few retailers to sell computers with 100 per cent Linux compatible hardware pre-loaded with Ubuntu. The desktop may not be Canonical's core business, but it is at System76—the end of the Unity project affected 91 per cent of its business. In response, it put together a distribution of its own to offer customers a user experience in line with the company's hardware. The company went distro hopping and settled on the latest Ubuntu with GNOME Shell as the base for its custom distro christened Pop!_OS.

Instead of focusing its efforts on mainstream desktop users, System76 is designing a distro geared towards creators, developers and makers: "If you're in software engineering, scientific computing, robotics, AI or IOT, we're building Pop!_OS for you," says CEO, Carl Richell in his blog post announcing the beta release. "We'll build tools to ease managing

your dev environment. We'll make sure CUDA is easily available and working."

System76 raised our expectations with its 'Imagine' marketing (<https://system76.com/pop>) that promises "the most productive and gorgeous platform for developing your next creation", even if it adds "we're just getting started". Speaking to Richell on the first full release, he explained that the company's intention is to establish the "guiding principles of the distribution, develop the aesthetic, build infrastructure, testing and quality procedures, start a community, documentation, web pages, production (shipping on laptops and desktops at scale) and of course, the release." Without this background, the first release seems bland and minimalistic given the objectives on its website. "In four months we've laid the infrastructure and direction of our OS. The next six months are



Above Sure, Pop is pitched at users who might not be aware of Gnome, but not mentioning the desktop even once might alienate those in the know

“As a good open source supporter, System76 is also improving upstream code**”**

executing on our vision”, explains Carl. We feel it’s important to bear this in mind when assessing the distro, but we can’t review what isn’t there yet.

On the software side, the distro is pretty standard desktop fare, with LibreOffice, Firefox and Gnome’s default bouquet of applications for viewing images and videos. Compared to Ubuntu, Pop uses Geary for email instead of Thunderbird and doesn’t include the default Ubuntugames, the pesky Amazon integration, Cheese, Transmission, Rhythmbox, and Shotwell. Carl says that in the company’s experience these missing apps aren’t widely used by customers.

Pop’s installation is based on Ubuntu’s Ubiquity installer. The only real difference apart from cosmetic ones is that there’s no initial user creation step. This has been moved to a post-install first-boot wizard. Pop’s GNOME desktop has been tweaked to deliver a user experience that best suits its customers. GNOME extensions are enabled that display workspaces whenever you bring up the Activities Overview, and another that adds a Suspend button in the Power menu. To protect your privacy, Pop also doesn’t display notifications on the lock screen. The System76 UI team has spent considerable time getting the visual details right.

The flat Pop GTK theme based on the Materia GTK+ theme with matching cursor and icons are pleasing.

As a good open source supporter, System76 is also improving upstream code. Carl tells us that it’s improved GNOME’s half-tiling window focusing and contributed patches to ensure HiDPI works properly. The one GNOME tweak that made things difficult for us were the remapped keyboard shortcuts. Carl tells us that the shortcuts have been worked out with input from software developers, but the tweaks necessitated a visit to the keyboard settings section to reassign them to the familiar GNOME values.

Another highlight is the Pop!_Shop app store, which is based on code from elementary OS’s AppCenter. While the distro uses the same repositories as Ubuntu 17.10, System76 handpicks the listed apps. This is why you can’t install apps such as Thunderbird, Evolution, Chromium from the Pop!_Shop. However, you can fetch them using APT. Another application that Pop has borrowed from the elementary OS project is Eddy, the DEB package installer. Carl tells us System76 are considering creating ‘application suites’ that will help users fetch multiple apps relevant to a particular task.

■ Mayank Sharma

Pros

An aesthetically pleasing GNOME rendition with useful extensions enabled by default and an intuitive app store.

Cons

A minimalist distro with a default application selection that so far fails to meet its grandiose objectives.

Summary

A good-looking distro that doesn’t yet match what it promises on the official website. But much of the work in this first release is in the background and lays the groundwork for future developments that’ll help push the distro as the perfect platform for developers.

7

CODE EDITOR

CudaText 1.23.0

A feature-rich editor for writing code



Linux has no dearth of advanced text editors that moonlight as lightweight IDEs as well and we ran a group test of some of the best a few issues ago

(LU&D182, Pg 81). CudaText is one such cross-platform editor that's primarily designed to write code, but can double up as an advanced text editor.

The editor has all the usual coding conveniences, like syntax highlight for several programming languages, including C, C++, Java, JavaScript, HTML, CSS, Python, and more. You also get code completion for some languages like HTML and CSS, code folding, the ability to search and replace with regex, as well multi-caret editing and multi-selection. You can extend the functionality by installing additional plugins that are written in Python.

The project releases precompiled binaries for Debian-based distros. On others you'll have to compile it manually by following the instructions on the wiki. It has a single document that explains all aspects of the app, and is a must-read to get to grips with all the functionalities of the app. If you've worked with advanced text/code editors before, you'll have no issues navigating CudaText's interface.

The screenshot shows the CudaText interface with a Python script named 'compute-hash.py' open. The code uses the hashlib module to compute the SHA-1 hash of a file. The interface includes a project tree on the left, a code editor in the center, and a plugin manager on the right. The plugin manager lists various Python-related plugins like 'Python_Fix_Imports' and 'Sort_By_Length'.

Above The app has colour themes for the interface. Each has a matching scheme for coding syntax

Pros

Includes all the common coding conveniences and can be extended with plugins.

Cons

You'll have to edit its configuration file to hook it up with the Python interpreter.

Great for...

Editing code without an IDE.
<http://uvviewsoft.com/cudatext/>

DIVE LOG TOOL

Subsurface 4.7.1

Log and analyse all your scuba dives with ease



Linus Torvalds likes to track whatever he does. He wrote Git to keep track of the kernel development and Subsurface to log his dives. Torvalds

likes to don the wetsuit and plunge underwater whenever he isn't busy hacking away at the Linux kernel. He couldn't find a good app to log his dives, so naturally he wrote one himself.

Simply put, Subsurface helps keep track of scuba dives. You can use it to import data from one of the supported dive computers, usually via USB. Once your data is imported, you can view and edit dive details from the intuitive user interface. The app also enables you to log dives manually. The app shows a dive profile graph that plots dives as a line graph.

It shows ascent and descent speeds along with the depths. Certain events, such as the begin of a decompression, are also marked. The app also records details about the diving equipment, and can calculate various stats about multiple dives.

There's a detailed user manual to help acquaint users, as well as a video tutorial. The latest version sports some user interface changes, like a new map widget. Support for importing dive data from Shearwater desktop, DL7, Datatrak and in other third-party formats has also been improved, and the version has experimental support for Bluetooth LE dive computers. The project lists binaries on its Downloads page for several popular distros, and there's an AppImage that'll work on any Linux distro.

Pros

Very easy to install and works with a large number of dive computers.

Cons

You'll have to read through its documentation to discover all its features.

Great for...

Analysis of dive data.
<https://subsurface-dive.log.org>

DESKTOP ENVIRONMENT

LXQt 0.12.0

Bring old workhorses to life with this lightweight desktop environment



The Lightweight Qt Desktop Environment—called LXQt for short—will draw a graphical user interface without consuming too many resources. The desktop environment is a mix of GTK-based lightweight desktop, LXDE, and Razor-Qt, which was an equally lightweight—but less mature—desktop that used the Qt toolkit.

The recent releases of the mainstream desktop environments such as Gnome and KDE have put them out of the reach of low-spec machines, which is why many popular distros have a LXQt-based flavour in their arsenal of releases. LXQt is also popular with users of newer more powerful machines, as it helps free the resources for more CPU-intensive tasks such as video processing.

The latest version of the desktop includes better support for HiDPI displays. The new release ships with a new Open/Save File dialog, and includes support for icon themes that use the `FollowsColorScheme` KDE extension to the XDG icon themes standard. Behind the scenes, the developers have also improved the shutdown/reboot process by shutting down all LXQt components before allowing `systemd` to do its job. There have been some important architectural changes too. The `lxqt-common` package has been dropped, and its components split into several other packages, such as the newly introduced `LXQt-themes` package.

The project's website has a list of distros that ship the LXQt desktop either in a spin or via their repositories.

Pros

A fully functional and smart-looking desktop that doesn't consume too many resources.

Cons

Doesn't include the bells and whistles you get with the mainstream desktop environments.

Great for...

Busy and old PCs.

<https://github.com/pixlra/playover>

IMAGE CONVERTER

Converseen 0.9.6.2

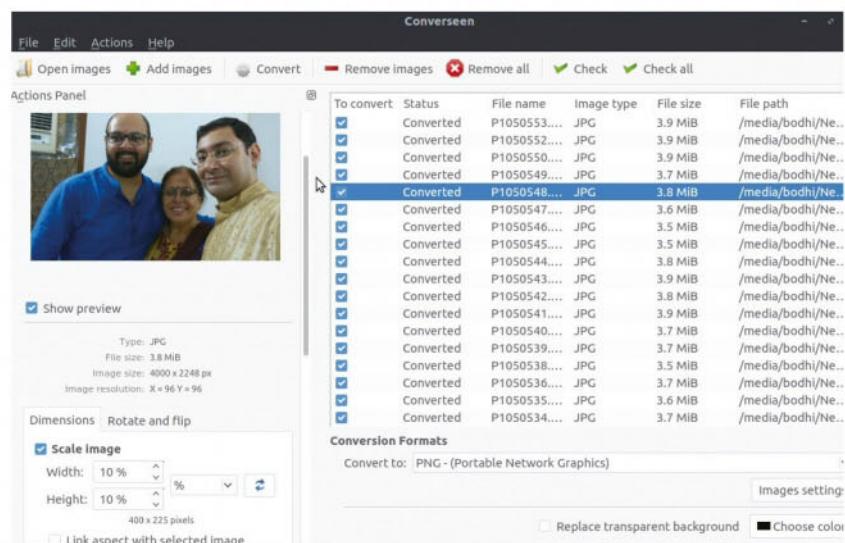
Automate repetitive image processing tasks



If you've worked with images, either professional or sorting through vacation clicks, you know a lot of image processing work is monotonous—repeatedly converting, resizing and rotating images to make them suitable for print or passing around. Although you can use virtually any image viewing or editing app, from DigiKam to GIMP, for this task, you'll save yourself hassle by employing a dedicated batch conversion tool like Converseen.

Converseen is a straightforward frontend to the command-line conversion utilities in ImageMagick. You can use the app to convert images to and from over 100 formats, rotate and flip them, change their dimensions and rename them in a fraction of the time it would take to perform these tasks manually.

You'll find installable binaries for Ubuntu, Fedora and OpenSUSE on the project's website, along with simple installation instructions. The workflow is pretty straightforward—click the Add Images button to select any number of images you want to convert. Then scroll down the Action panel on the left to specify how you want to manipulate the image, and hit the Convert button to begin the process.



Above Converseen exposes only a fraction of the image manipulation tricks you can do with ImageMagick

Pros

Straightforward interface that helps automate mundane image editing tasks

Cons

Exposes a very limited subset of the power of the CLI tools it's based on.

Great for...

Resizing, rescaling and rotating.
<http://converseen.fasterland.net>

RECOMMENDED

Hosting listings

Featured host:

www.thenames.co.uk
0370 321 2027



About us

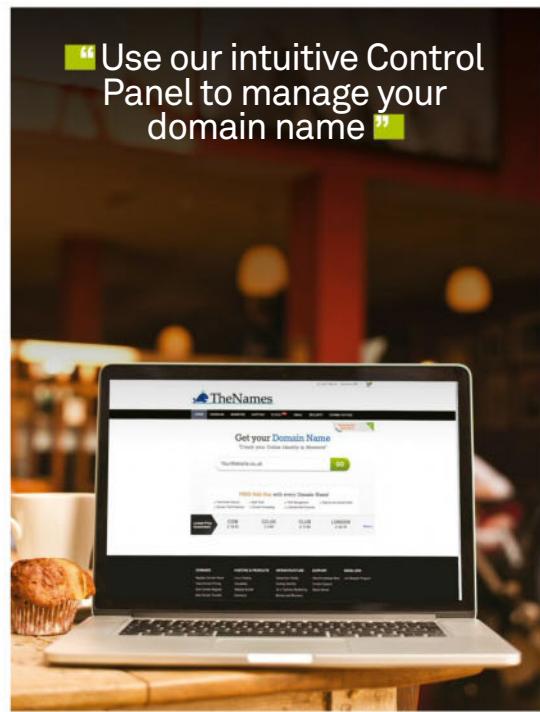
Part of a hosting brand started in 1999, we're well established, UK based, independent and our mission is simple – ensure your web presence 'just works'.

We offer great-value domain names, cPanel web hosting, SSL certificates, business email, WordPress hosting, cloud and VPS.

What we offer

- Free email accounts with fraud, spam and virus protection.
- Free DNS management.
- Easy-to-use Control Panel.
- Free email forwards – automatically redirect your email to existing accounts.
- Domain theft protection to prevent it

- being transferred out accidentally or without your permission.
- Easy-to-use bulk tools to help you register, renew, transfer and make other changes to several domain names in a single step.
- Free domain forwarding to point your domain name to another website.



5 Tips from the pros

01 Optimise your website images

When uploading your website to the internet, make sure all of your images are optimised for the web! Try using jpegmini.com software; or if using WordPress, install the EWWW Image Optimizer plugin.

02 Host your website in the UK

Make sure your website is hosted in the UK, not just for legal reasons! If your server is located overseas, you may be missing out on search engine rankings on google.co.uk – you can check where your site is based on www.check-host.net.

03 Do you make regular backups?

How would it affect your business if you lost your website today? It's vital to always make your own backups; even if

your host offers you a backup solution, it's important to take responsibility for your own data and protect it.

04 Trying to rank on Google?

Google made some changes in 2015. If you're struggling to rank on Google, make sure that your website is mobile-responsive! Plus, Google now prefers secure (HTTPS) websites! Contact your host to set up and force HTTPS on your website.

05 Avoid cheap hosting

We're sure you've seen those TV adverts for domain and hosting for £1! Think about the logic... for £1, how many clients will be jam-packed onto that server? Surely they would use cheap £20 drives rather than £1k+ enterprise SSDs! Remember: you do get what you pay for!

Testimonials

David Brewer

"I bought an SSL certificate. Purchasing is painless, and only takes a few minutes. My difficulty is installing the certificate, which is something I can never do. However, I simply raise a trouble ticket and the support team are quickly on the case. Within ten minutes I hear from the certificate signing authority, and approve. The support team then installed the certificate for me."

Tracy Hops

"We have several servers from TheNames and the network connectivity is top-notch – great uptime and speed is never an issue. Tech support is knowledge and quick in replying – which is a bonus. We would highly recommend TheNames."

J Edwards

"After trying out lots of other hosting companies, you seem to have the best customer service by a long way, and all the features I need. Shared hosting is very fast, and the control panel is comprehensive..."

Supreme hosting



www.cwcs.co.uk
0800 1 777 000

CWCS Managed Hosting is the UK's leading hosting specialist. It offers a fully comprehensive range of hosting products, services and support. Its highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and passionate about what they do.

- Colocation hosting
- VPS
- 100% Network uptime

Value hosting

elastichosts

elastichosts.co.uk
02071 838250

ElasticHosts offers simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Its team of engineers provide excellent support around the clock over the phone, email and ticketing system.

- Cloud servers on any OS
- Linux OS containers
- World-class 24/7 support

Small business host



www.hostpapa.co.uk
0800 051 7126

HostPapa is an award-winning web hosting service and a leader in green hosting. It offers one of the most fully featured hosting packages on the market, along with 24/7 customer support, learning resources, as well as outstanding reliability.

- Website builder
- Budget prices
- Unlimited databases



Enterprise hosting:



www.2020media.com | 0800 035 6364

WordPress comes pre-installed for new users or with free managed migration. The managed WordPress service is completely free for the first year.

We are known for our 'Knowledgeable and excellent service' and we serve agencies, designers, developers and small businesses across the UK.



Budget hosting:



www.hetzner.de/us | +49 (0)9831 5050

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997 the company has provided private and business clients with high-performance hosting products, as well as the necessary infrastructure for the efficient operation of websites. A combination of stable technology, attractive

pricing and flexible support and services has enabled Hetzner Online to continuously strengthen its market position both nationally and internationally.

- Dedicated and shared hosting
- Colocation racks
- Internet domains and SSL certificates
- Storage boxes

SSD web hosting



www.bargainhost.co.uk
0843 289 2681

Since 2001, Bargain Host has campaigned to offer the lowest possible priced hosting in the UK. It has achieved this goal successfully and built up a large client database which includes many repeat customers. It has also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting



patchman-hosting.co.uk
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven ecommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

Fast, reliable hosting



www.bytemark.co.uk
01904 890 890

Founded in 2002, Bytemark are "the UK experts in cloud & dedicated hosting". Their manifesto includes in-house expertise, transparent pricing, free software support, keeping promises made by support staff and top-quality hosting hardware at fair prices.

- Managed hosting
- UK cloud hosting
- Linux hosting

Free Gifts



WHAT IS IT?

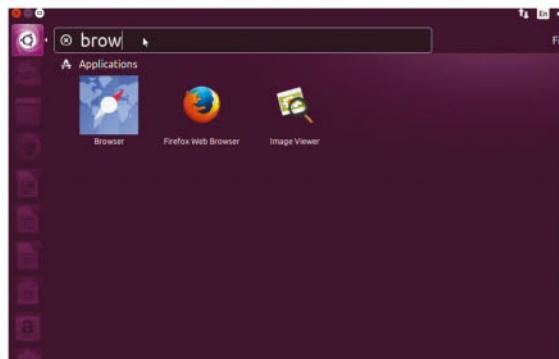
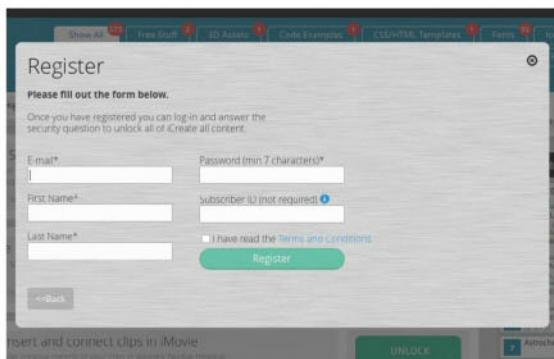
Every time you see this symbol in the magazine, there is free online content that's waiting to be unlocked on FileSilo.

WHY REGISTER?

- Secure and safe online access, from anywhere
- Free access for every reader, print and digital
- Download only the files you want, when you want
- All your gifts, from all your issues, all in one place

Get your free resources

Download the best distros, essential FOSS and all our tutorial project files from your FileSilo account



1. UNLOCK YOUR CONTENT

Go to www.filesilo.co.uk/linuxuser and follow the instructions on screen to create an account with our secure FileSilo system. When your issue arrives or you download your digital edition, log into your account and unlock individual issues by answering a simple question based on the pages of the magazine for instant access to the extras. Simple!

2. ENJOY THE RESOURCES

You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices. If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at filesilohelp@futurenet.com.

Free for digital readers too! Read on your tablet, download on your computer

Most Downloaded Files

1. Caffeine Tutorial File (1.0M)
2. Key tutorial-Tutorial File (3.7M)
3. Creating dynamic templates with Hack, Java2 and Twitter-Tutorial File (2.0M)
4. Build an XBMC extension with Python-Tutorial File (2.0M)
5. Build a complex LED matrix-Tutorial File (2.0M)
6. Make a Breakout clone with Pygame Zero-Tutorial File (2.0M)
7. Twitter OAuth File-Tutorial File (1.0M)

→ Log in to www.filesilo.co.uk/linuxuser



Subscribe and get instant access

Get access to our entire library of resources with a money-saving subscription to the magazine – subscribe today!

This month find...

DISTROS

In the FileSilo this month you will find Antergos 17.11, an easy-to-install distro powered by Arch Linux, along with Xubuntu 17.10 an optimised Ubuntu distro.

SOFTWARE

As well as two full distros, we've bundled together a Rescue and Repair kit that includes two popular live distros, SystemRescueCd and Rescatux, that will help you bring a sick system back to life.

TUTORIAL CODE

This month we've got the skeleton project for MQTT, Python code for Arduino and a TAR for the Java series and more!



Subscribe & save!

See all the details on how to subscribe on **page 30**

FOLLOW US



Facebook:

facebook.com/LinuxUserUK



Twitter:

@linuxusermag

NEAR-FUTURE FICTION

Disjointed



ABOUT

Eating Robots

Taken from the new book *Eating Robots* by Stephen Oram: near-future science-fiction exploring the collision of utopian dreams and twisted realities as humanity and technology become ever more intertwined. Sometimes funny and often unsettling, these 30 sci-fi shorts will stay with you long after you've turned the final page.

<http://stephenoram.net>

He stood naked in front of the full-length mirror, flexing his biceps. The mirror flashed an amber warning, reminding him to stand still while it scanned his organs, blood, bones and skin. It would evaluate his health and adjust the multitude of enhancement implants scattered throughout his body, fine-tuning them as it went to maximise his physical and mental performance.

This daily routine made him feel trapped and cornered, as if the mirror was a docking bay that he couldn't stay away from for more than twenty-four hours. He wanted to run into the sea or drink himself silly. He wanted to go off-grid and wander wherever he liked and for as long as he liked.

The pressure to break free had been building for a while, to such an extent that he doubted whether he could make it through another day.

As the timer in the top right-hand corner approached zero, he tightened his stomach muscles and straightened his back. The mirror snapped its daily photo for his archives and, he suspected, a central database.

He clenched his fist and punched the mirror with all the force he could summon. A thousand pieces flew across the room. The holistic guardian of his well-being was dismembered and lying scattered all around him.

An enormous sense of relief welled up from deep inside.

But.

What was that?

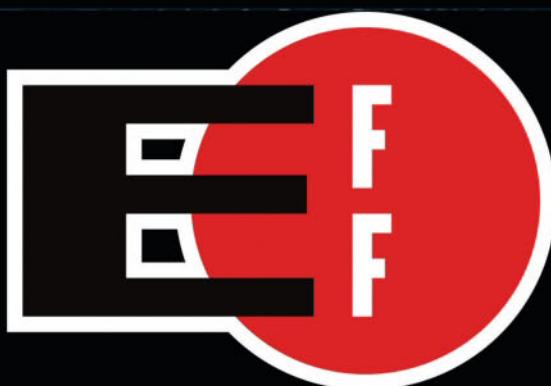
The fragments of the mirror were continuing their work in isolation and different parts of his body were choosing their own settings.

His hands were getting warmer, his feet colder. His heart was racing. His stomach clenched and his calf muscles cramped. And yet as soon as his brain registered a problem, it told him not to worry, immediately overriding all warnings.

He screamed as the pain and euphoria of the dissection reached every part of his being. ■

00.00

Good morning, Who do you want to be today?



The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier

LOSE YOURSELF IN A WORLD OF

Vinyl

FIND YOURSELF IN
OXFAM'S ONLINE SHOP

oxfam.org.uk/shop



OXFAM