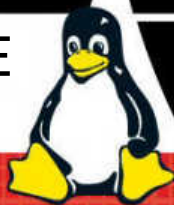


FREE TUTORIAL FILES, VIDEOS & FOSS PACKAGES

www.linuxuser.co.uk

LinuxUser

THE ESSENTIAL MAGAZINE
FOR THE GNU GENERATION



& Developer™

GO WIRELESS!

25 MAKER PROJECTS FOR RASPBERRY PI 3

GREAT NEW IDEAS FOR
WI-FI & BLUETOOTH



**CLOUD
STORAGE
TESTED**

**15
NEXT-GEN
DISTROS**

Innovative distributions
that change everything

**“ENCRYPTION IS KEY
TO A HEALTHY INTERNET”**
Mark Surman, Mozilla Foundation

ip **DigitalEdition**
GreatDigitalMags.com
ISSUE 164

**MASTER NEW
SENSE HAT SKILLS**
Build a selection interface
and display images



PLUS TOR ROUTER, MATTERMOST & DNS PROXY



RPI2

GOOD FROM ANY ANGLE

Imagine Publishing Ltd
Richmond House, 33 Richmond Hill
Bournemouth, Dorset, BH2 6EZ
☎ +44 (0) 1202 586200
Web: www.imagine-publishing.co.uk
www.linuxuser.co.uk
www.greatdigitalmags.com

Magazine team

Editor Gavin Thomas

gavin.thomas@imagine-publishing.co.uk
☎ 01202 586257

Production Editor Rebecca Richards

Features Editor Oliver Hill

Designer Sam Ribbits

Photographer James Sheppard

Senior Art Editor Andy Downes

Editor in Chief Dan Hutchinson

Publishing Director Aaron Asadi

Head of Design Ross Andrews

Contributors

Dan Aldred, Joey Bernard, Christian Cawley, Kunal Deo, Gareth Halfacree, Tam Hanna, Richard Hillesley, Jon Masters, Ashish Sinha, Richard Smedley, Mark Surman, Nitish Tiwari, Mihalis Tsoukalos

Advertising

Digital or printed media packs are available on request.

Head of Sales Hang Deretz

☎ 01202 586442

hang.deretz@imagine-publishing.co.uk

Sales Executive Luke Biddiscombe

☎ 01202 586431

luke.biddiscombe@imagine-publishing.co.uk

FileSilo.co.uk

Assets and resource files for this magazine can now be found on this website.

Support filesilohelp@imagine-publishing.co.uk

International

Linux User & Developer is available for licensing.

Head of International Licensing Cathy Blackman

☎ +44 (0) 1202 586401

licensing@imagine-publishing.co.uk

Subscriptions

For all subscriptions enquiries

☎ 0844 249 0282 (UK)

☎ +44 (0) 1795 418661 (Overseas)

Email: LUD@servicehelpline.co.uk

6 issue subscription (UK) – £25.15

13 issue subscription (Europe) – £70 (ROW) – £80

Circulation

Head of Circulation Darren Pearce

☎ 01202 586200

Production

Production Director Jane Hawkins

☎ 01202 586200

Finance

Finance Director Marco Peroni

Founder

Group Managing Director Damian Butt

Printing & Distribution

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

Distributed in the UK, Eire & the Rest of the World by:
Marketforce, 5 Churchill Place, Canary Wharf
London, E14 5HU
☎ 0203 148 3300
www.marketforce.co.uk

Distributed in Australia by:

Gordon & Gotch Australia Pty Ltd
26 Rodborough Road
Frenchs Forest, New South Wales 2086, Australia
☎ +61 2 9972 8800
www.gordongotch.com.au

Disclaimer

The publisher cannot accept responsibility for any unsolicited material lost or damaged in the post. All text and layout is the copyright of Imagine Publishing Ltd. Nothing in this magazine may be reproduced in whole or part without the written permission of the publisher. All copyrights are recognised and used specifically for the purpose of criticism and review. Although the magazine has endeavoured to ensure all information is correct at time of print, prices and availability may change. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to Imagine Publishing via post, email, social network or any other means, you automatically grant Imagine Publishing an irrevocable, perpetual, royalty-free license to use the material across its entire portfolio, in print, online and digital, and to deliver the material to existing and future clients, including but not limited to international licensees for reproduction in international, licensed editions of Imagine products. Any material you submit is sent at your risk and, although every care is taken, neither Imagine Publishing nor its employees, agents or subcontractors shall be liable for the loss or damage.

Welcome

to issue 164 of Linux User & Developer

This issue

- » Discover 25 projects made possible with Pi 3
- » Test out the next generation of distros
- » Create a simple programming language
- » Set up a DNS proxy and Tor router



Welcome to the latest issue of Linux User & Developer, the UK and America's favourite Linux and open source magazine. Change is upon us! Wi-Fi and Bluetooth modules finally came within reach of the Raspberry Pi Foundation's \$35 price point for the Model B, so the recently released Pi 3 ships with excellent wireless capabilities – not to mention a 1.2 GHz, 64-bit quad-core

ARM CPU. James Adams, the Foundation's director of hardware, explains all from page 18. Turn to page 22 to find the projects we've come up with, and see page 26 for Gareth Halfacree's full review.

Taking a wider view of the Linux landscape, we've also picked out 15 of the very best distributions that we consider to be next-gen OSs. These are distros that reinvent the UX, refresh the UI, rework the core system, and otherwise reinvigorate a community that often seems to be dominated by the same handful of long-established distros. Flip over to page 18 to find them.

We've also reached the end of two tutorial series this month: MonoGame and computer science, with a look at advanced shader code and how to write a custom programming language, respectively. Plus, in the Pi section we have some great guides to using the Sense HAT and setting up a Tor router, as well as a Fritzing masterclass. Enjoy the new issue!

Gavin Thomas, Editor

Get in touch with the team:

linuxuser@imagine-publishing.co.uk



Buy online
imagineshop.co.uk

Visit us online for more news, opinion, tutorials and reviews:

www.linuxuser.co.uk

Contents

Subscribe & save!

28 Check out our great new offer! US customers can subscribe via page 86

30 Next-Gen Distros

Discover 15 groundbreaking Linux distributions

OpenSource

- 06 News**
The biggest stories from the open source world
- 08 Free software column**
Expert insight into open source and free software
- 11 Coding column**
Learn problem solving and systems programming in C
- 12 Interview**
Mozilla's Mark Surman talks about encryption education
- 16 Kernel column**
The latest on the Linux kernel with Jon Masters

Features

- 18 Raspberry Pi 3**
25 New projects to try and an interview with James Adams
- 30 Next-gen distros**
See what's in store for the future of Linux systems

Tutorials

- 40 Systems programming: Advanced file input & output**
Master multiplexing, record locking and more
- 46 Access Hadoop data with HBase**
Easily process massive sets of data
- 50 Advanced shading in MonoGame**
Finish up your first 3D project by adding in bump mapping, post-processing and more
- 54 Set up your own DNS proxy server**
Boost privacy by obscuring your IP address
- 58 Simplify team communication with Mattermost**
Start using a state-of-the-art chat platform
- 62 Computer science: Create a programming language**
Put your computing skills to the test!

Resources



96 Free downloads

Find out what we've uploaded to our digital content hub FileSilo for you this month

Reviews

87 Best cloud storage

What is the best service for storing and syncing your files and media?



ownCloud



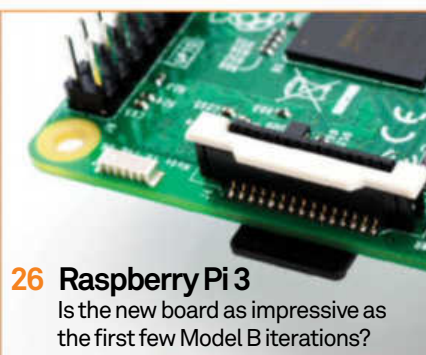
Bitcasa



Cozy



Yandex.Disk

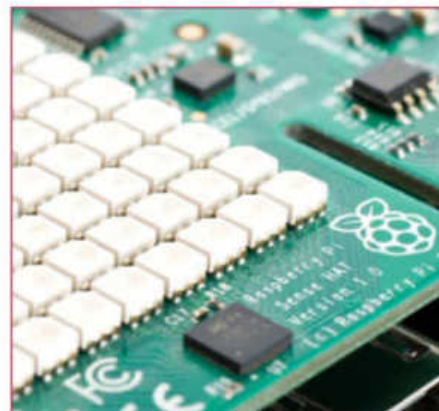


26 Raspberry Pi 3

Is the new board as impressive as the first few Model B iterations?

92 Free software

Richard Smedley recommends some excellent FOSS packages for you to try



69 Practical Raspberry Pi

Learn how to handle 2D and 3D graphics using OpenGL in Python, practice essential Sense HAT skills and start coding a game, make a Tor router, and master Fritzing.

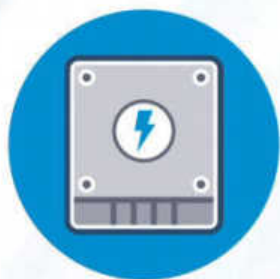
Join us online for more Linux news, opinion and reviews www.linuxuser.co.uk

Dedicated Servers

Get 1Gbit/s
connectivity



Unlimited monthly
data transfer



Smart SSD technology



UK data centres
and 24/7 support

NEW

Dedicated Servers from:

£29.00
per month

£29.00 ex VAT for 3 months then £39.00 ex VAT per month.
12 month minimum term contract. One-off set up of £49.00 ex
VAT applies. 6 months discount available on selected servers.
See website for terms and conditions

Call **0808 1686 777**
or visit **fasthosts.co.uk**

SERVICES • WEB HOSTING • DOMAIN NAMES • EXCHANGE EMAIL



WEARABLES

AsteroidOS: Wear an open source watch



Unlock the potential of your Android Wear smartwatch with a new, libre operating system

For a long time, Android wear has led the way when it comes to throwing the spotlight onto the world of smartwatches. What was initially perceived as a bit of a gimmick from early adopters has now turned into a series of invaluable devices that are essential work and personal aides for people around the world. Advancements in hardware and software have led to a swathe of features that would primarily have been restricted to our smartphones all becoming possible on our wrists.

But, until now, the open source community has largely been left behind when it comes to the world of smartwatch operating systems. With Google and Apple leading the charge in this field, there's been little competition, and any independent launches have been heavily restricted in what they can do. However, we now finally have a choice with the launch of AsteroidOS.

Developed by Florent Revest, AsteroidOS is the first true open source distribution that's been specifically made for smartwatches. In its current state, AsteroidOS contains only the bare minimum of functions. So things like a calculator, stopwatch and calendar are all ready to use, but many of the smartphone connection features we've become accustomed to are not.

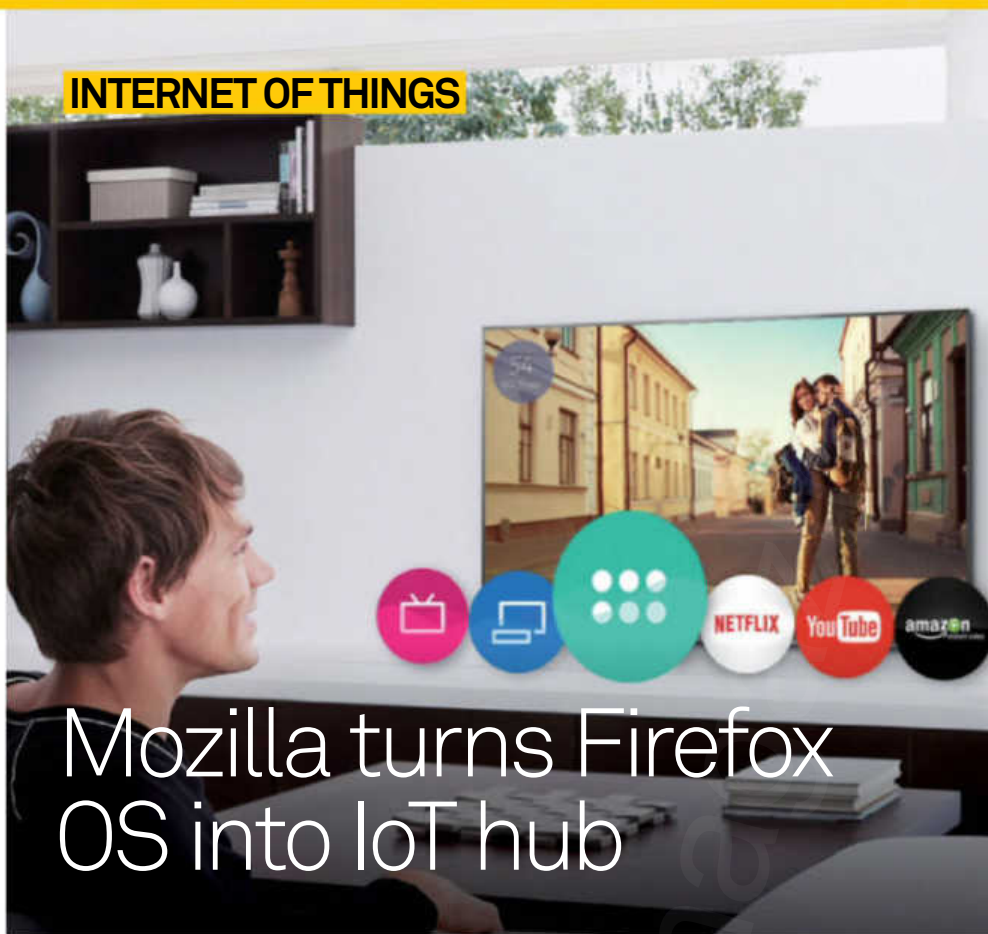
Adoption rates are still in their infancy, and Revest has been quick to state that the project will move at a much faster speed once more developers are working on it. Currently, owners of the LG G Watch and LG G Watch Urbane have full access to the OS, but as long as your bootloader is unlocked, porting it over to other devices isn't a difficult task. The kernel source is built around libhybris, which allows the re-use of Android drivers.

Initial tests on the AsteroidOS software have proved positive. On videos uploaded to

AsteroidOS' official Twitter account, videos have shown off many of its core apps working. Speeds are quick, crashes are infrequent and there's plenty of on-board customisation on hand. Keen developers can go hands-on with the code that Revest and his team have uploaded to GitHub. There's scope to build applications from scratch and directly implement them on your smartwatch, and even change the way that your hardware performs. Due to the open source nature of the operating system, users also have the tools to adopt the OS for use on various other wearables. So perhaps a VR version of AsteroidOS could be with us down the line.

Visit the official AsteroidOS website (asteroidos.org) for a full overview of the project, where you'll find a guide to installing the OS on to your device, and also GitHub where you can check out its source code for yourself.

INTERNET OF THINGS



Mozilla turns Firefox OS into IoT hub

Firefox OS is transformed to bring IoT into our lives

As an operating system, Firefox has undergone a massive transformation in the past 24 months – it's far more than just a web browser nowadays. But now Mozilla are looking to take Firefox to the next level by using it as a hub for a plethora of Internet of Things projects.

Mozilla is currently working on four IoT projects behind the scenes: Project Smart Home, Project Link, Project Sensor Web and Vaani. Each of the projects will look to deal with IoT technology in different ways, but all are aimed at making the end consumer's home and devices smarter. In a blog post, Mozilla's SVP of Connected Devices, Ari Jaaksi, posted: "Everything is connected around us. This revolution has already started and it will be bigger than previous technology revolutions, including the mobile smartphone revolution. Internet of Things, as many call it today, will fundamentally affect all of us."

Smart Home is pretty self-explanatory, and looks to give users the tools needed to automate some of their home's basic functions, while still giving more discerning users some

useful flexibility in how they adopt it into their home. With Link, users will have a new personal assistant that bridges the gap between themselves and their smart devices. Over prolonged time, Link will remember certain preferences, making it faster and eventually requiring minimal user interaction.

Vaani is the name given to Mozilla's new voice interface that can be integrated into third-party apps and hardware. To start with, Vaani will recognise simple commands, with a view to catering to more complex statements down the line. Sensor Web is a little different, and relies on a network of sensors to collate crowd-sourced data that will be openly accessible to all. Mozilla will process the data and present it to users to view in their own time.

Many of Mozilla's outlined projects are very much in their early phases and it's not known when we'll see any of these actively in our homes. With Mozilla's continued development of the Firefox platform, it'll be interesting to see where its new IoT focus leads it.

TOP FIVE

New features in Android N

1 Multi-window mode

One of the most hotly anticipated features of N is its new multi-window mode. With this enabled, users can seamlessly switch between multiple apps in seconds, or when needed, use multiple apps simultaneously.

2 Interactive notifications

In a move that closely resembles iOS, N users will now be able to reply to messages directly from their notification panel. A quick overview of the message will be shown with users needing to simply double tap on it to start a reply. You'll never need the Messaging app again!

3 Souped-up settings

An overarching system menu has been implemented to help get around the Settings app at a much quicker pace. No matter what menu you're in, a press of the recognisable menu icon will show an overview of the core settings, which can then be edited depending on what you need from it.

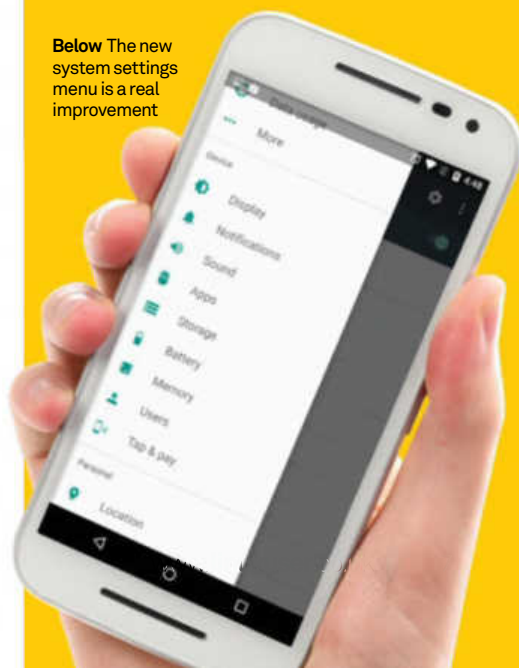
4 Enhanced Doze mode

Doze was a great way of saving battery, but it was limited in its functionality. A new and improved Doze mode has been implemented into N, which allows for greater control over your device's battery life, as well as more accurate tracking.

5 Android beta program

Users can now opt-in to a new Android beta program on their devices. This allows Google to transmit preview builds and minor updates OTA, without the need for flashing factory images.

Below The new system settings menu is a real improvement



OPINION FREE SOFTWARE

The Joy of Sun and ZFS

‘GPLv2 incompatibility was Sun’s design choice’ – so what does this mean for Canonical’s use of ZFS in Ubuntu 16.04?



Canonical’s decision to include ZFS in Ubuntu 16.04 LTS has become the source of some controversy. ZFS is

the open source File System and Logical Volume Manager introduced with OpenSolaris in late 2005, and was released under Sun Microsystems’s Common Development and Distribution License (CDDL), which the Free Software Foundation has always described as incompatible with the GPLv2, thus rendering ZFS inaccessible to GNU/Linux.

Until its purchase by Oracle, Sun Microsystems had a considerable history as a contributor and beneficiary of open source projects. Bill Joy, the prime architect of SunOS and Solaris, was the lead developer of BSD Unix, and wrote vi, NFS, and Berkeley’s TCP/IP stack before moving to Sun. Sun’s support of free and open source software was sometimes ambivalent but often significant, as its support of Java, the GNOME desktop, Mozilla and OpenOffice.org and countless other projects demonstrated, but there was always a conflict between its open source heritage and the internal and external pressures it faced to retain control of its “IP”, so Sun always retained ownership and tight control of its projects through ownership of patents, in the case of Java, and devices such as copyright assignment, in the case of OpenOffice.org.

ZFS was licensed under the CDDL at a time when the OSI encouraged and accepted a proliferation of open source licences. CDDL was perceived as a quick and desirable option by Sun. Software Freedom Conservancy assert that “We believe Sun was aware when drafting CDDLv1 of the incompatibilities; in fact, our research into its history indicates the GPLv2-incompatibility was Sun’s design choice. At the time, Sun’s apparent goal was to draw developers away from GNU and Linux development into Solaris. Not only did Sun not want code from GNU and Linux in Solaris, more importantly, Sun did not want technological advantages from Solaris’ kernel to appear in Linux.”

Incompatibility with the GPL may have been deliberate, but times have changed. Sun’s offerings have long been subsumed by Oracle, and Oracle



Richard Hillesley

writes about art, music, digital rights, Linux and free software for a variety of publications

“Software Freedom Conservancy and their lawyers are unequivocal in their belief that Canonical’s inclusion of ZFS in an Ubuntu distribution is non-compliant with the terms of the GPL”

distributes its own version of Linux, a clone of Red Hat Enterprise, and could resolve the licensing issue by releasing ZFS under a less ambivalent licence, but Oracle is first and foremost a proprietary software company, and has little interest in making life easier for the free and open source software community.

Software Freedom Conservancy and their lawyers are unequivocal in their belief that Canonical’s inclusion of ZFS in an Ubuntu distribution is non-compliant with the terms of the GPL. By way of contrast, Eben Moglen and the Software Freedom Law Center take a contrary view, justified by exceptions and statements by Linus Torvalds and the kernel COPYING file that allow for a Linux-specific definition of what is and is not considered to be a ‘derivative work’.

“ZFS has been made available by programmers working from the free software CDDL-licensed versions,” wrote Moglen, “for use as a ‘userspace filesystem’ in GNU/Linux systems through the FUSE APIs. This mode of use presents no licensing issues: the filesystem is running as a distinct userspace process making system calls to the kernel through stable, defined, standard interfaces. As the kernel COPYING file states, in that mode of employment the filesystem is not a ‘derived work’ of the kernel, which means it is not within the scope of the kernel copyleft.”

This is not a unanimous view among lawyers, kernel developers and the individual owners of copyrighted kernel code, so the argument may run for some time. Exceptions may be seen to weaken the definition of a “derived work” and reduce the effects of copyleft. On the other hand, Torvalds is clear that “anything that was written with Linux in mind (whether it then also works on other operating systems or not) is clearly partially a derived work,” and “anything that has knowledge of and plays with fundamental internal Linux behaviour is clearly a derived work. If you need to muck around with core code, you’re derived, no question about it.”

How the law in a European jurisdiction interprets a ‘derived work’ may become clearer when the case between Christoph Hellwig and VMware is resolved over the coming weeks and months. ■

OPEN HARDWARE

Google has joined Facebook's Open Compute Project

Two of tech's biggest players want to revolutionise the world of data centres

Facebook's Open Compute Project has been picking up a lot of steam as of late, and the acquisition of Google shows its momentum.

The project was originally created six years ago for end-user companies to collaborate on data centre equipment, as a way of keeping costs down and free of superfluous features.

While a number of other technology giants have already jumped on board with the project, including the likes of Microsoft, Google has long been absent from the list until the recent OCP Summit in Silicon Valley.

Google is already touted for owning some of the world's most advanced data centres, and its first contribution will be to produce a new rack design that powers servers with an enhanced 48-volt current, compared to the standard 12 volts used in most data centres currently. It's believed that the technology behind Google's radical take on rack design is ready to be deployed for worldwide use.

"The key thing we figured out was, to get the efficiency in cost and power, you've to directly feed the 48 volts to the motherboard

and convert it only in one step," said Urs Holzle, SVP of Google's infrastructure, at the recent OCP Summit. "The OCP's current rack design is too deep for the narrow aisles in Google's data centres."

The new design is a partnership between Google and Facebook, with both trialling the design in a small number of their own global data centres. Google's involvement leaves Amazon as the only hyperscale cloud provider that isn't a member of the OCP, following Apple's membership near the end of last year.

GAMING

MAME is now free and open source

Arcade machine emulator now free to be modified

After nearly two decades of availability, MAME (Multiple Arcade Machine Emulator) has finally made its transition across to open source availability. Previously the software was available under a BSD licence, which prohibited any sort of commercial use of MAME.

Despite the legal grey areas that MAME has had to deal with in its lifespan, it still remains far and away the most popular way for gaming enthusiasts to emulate and enjoy retro titles from yesteryear.

Up until recently, the source code for the MAME software had been readily available, but heavy restrictions were in place to limit what outside users were able to do. Users can now modify and distribute MAME to their heart's content. It'll be interesting to see where the project ends up, and how it's integrated into other projects.

"After 19 years, MAME is now available under an OSI-compliant and FSF-approved license," stated the team over on mamedev.org: "A great majority of files (over 90% including core files) are available under the 3-Clause BSD license, but the project as a whole is distributed under the terms of the GNU

General Public License." The transition into a completely open source project hasn't exactly been easy, with the team also stating that the past ten months have been filled with trying to contact all the people who have contributed to MAME over its lifespan with information about the new licence. If you believe that you've contributed to the core MAME source code, the MAME team has included a link where you can reach out to them directly over on their site.

Right ROMs from classic arcade machines like this can be run inside the MAME software



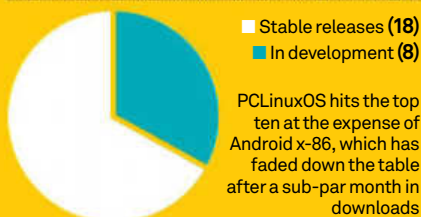
DISTRO FEED

► Top 10

(Average hits per day, 19 February – 16 March)

1.	Linux Mint	▲	3,027
2.	Debian	▼	2,043
3.	Ubuntu	▲	1,943
4.	openSUSE	▼	1,361
5.	Manjaro	▲	1,299
6.	Fedora	▲	1,176
7.	PCLinuxOS	▲	1,035
8.	Zorin	▼	994
9.	CentOS	▲	907
10.	Arch	▲	889

► This month



► Highlights



Red Hat 6.8 Beta

The new RHEL beta gives customers a chance to preview some of the new features, including the likes of Relax-and-Recover archiving and the newly implemented Libreswan VPN software.



Manjaro

Manjaro now provides a completely new graphical interface, making it easier than ever to install and remove the many kernels they offer. Its selection of kernels is unrivalled by its competition.



Untangle NG

There's a new look to Untangle's firewall service, making it easier for new users to customise their desktop for security. No core features have been altered, so seasoned users will be able to jump straight back in.

Latest distros available:

filesilo.co.uk



FOSS

Writer upgrades shine in LibreOffice 5.1.1 update

A much-requested feature is now readily available

It seems like only yesterday that we were looking at the 5.1.0 release of the ever-popular LibreOffice suite, but the team behind it has recently announced the first point update for the 5.0 series. "The Document Foundation releases LibreOffice 5.1.1, the first minor release of the LibreOffice 5.1 family," said Italo Vignoli. "The latest update is targeted at technology enthusiasts, early adopters and power users."

While the latest update mainly irons out many of the bugs that troubled the previous

instalment, there are also a number of new features worth getting excited about. The software now has vastly deeper integration with cloud servers, enabling users to not only read documents stored in accounts, but also make subsequent edits to them. Transitions to the cloud seem generally quicker and easier to complete based on initial feedback.

Arguably the biggest changes appear in Writer, with users now able to hide the white space normally found between pages, which helps to create a continuous flow of text and makes documents much more readable, as well as neatly handling awkward breaks created by compatibility issues with proprietary document formats.

Along with these new changes, there's also an outline split button now available in the formatting toolbar and a new jump feature within print preview, to help you identify an individual page for printing. A full documentation of the new features and edits can be found alongside a download of the official LibreOffice client over at www.libreoffice.org.



Above You can now simplify documents for reading by hiding white space between pages

DATABASE

Microsoft's opens up SQL Server

A benchmark for data management and analytics

Microsoft's much-rumoured ventures into open source software have been made clearer after its official announcement that its popular SQL Server will be made fully available on Linux desktops.

In a report published by Microsoft's Executive VP, Cloud and Enterprise Group, Scott Guthrie, SQL Server capabilities on Linux will "deliver a consistent data platform across Windows servers and Linux, as well as on-premises and cloud. SQL Server on Linux will provide customers with even more flexibility in their data solution."

Linux users will have the complete suite of SQL Server features available to them by default, including the likes of Stretch Database, which acts as a low-cost solution for customers to access their data whenever they need to. The announcement will now mean that developers can now build applications based around SQL's superb enterprise capabilities.

The preview of SQL Server for Linux is now available, with access to its suite of core database capabilities included. Full availability has been slated for release in mid-2017.

Counting string length

This month we explore the most popular string-related C function, `strlen()`, so we can measure and traverse strings easily

“The `strlen()` function finds the length of a string by counting the number of characters that precede the terminating NUL character

Every C string is required to be stored in this way; if there is no terminating character, C cannot determine where a string ends – a nasty bug that the compiler cannot detect!

The most important thing to remember is that a string in C is terminated with the `'\0'` character. Do not get confused; in C, `'\0'` is a single character despite the fact that it needs two characters to get printed on your screen.

As always, there are two ways to deal with strings in C: using pointers or arrays. Similarly, you can use a for loop as well as a while statement to process a string character by character – picking one of them is a matter of taste.

Even if you define a C string using an array, the C string can be accessed using pointers and vice versa. As the terminating character occupies a single memory space, a string with five characters takes six memory spaces: five for the string itself and one for the terminating character. If you use static strings, C allocates the additional memory space and puts the terminating character:

```
char * aString = "12345";
char anotherString[] = "12345";
```

If you decide to allocate memory space for a string using `malloc()` you should manually include the additional space for the terminating character, as in the next example:

```
char *aThirdString = malloc(strlen("12345") + 1);
```

This is one of the most frequent uses of `strlen()`. To make things more interesting, the pointer version of the `strlen()` implementation (`strlenPointer.c`) will use a while statement whereas the array version (`strlenArray.c`) will use a for loop. Once again, the input string will be given as a command line argument in both versions of the program. Last, in order to avoid confusions and confusions with the C `strlen()` function, the developed function will be named `myStrlen()` but it will take the same input and produce the same output as the standard `strlen()` function – please bear in mind that `strlen()` returns a `size_t` type of variable, which is an unsigned integer type of at least 16 bits. The fact that `size_t` is an unsigned type means that it can be used for counting



Mihalis Tsoukalos

is a UNIX administrator, a programmer, a DBA and a mathematician. He has used Linux since 1993

“Even if you define a C string using an array, the C string can be accessed using pointers and vice versa”

positive values, such as the length of a string, but it cannot be used for representing negative quantities. The correct `printf()` placeholder for printing `size_t` values is `%zu` – use it to avoid compiler warnings like this:

warning: format ‘%d’ expects argument of type ‘int’, but argument 3 has type ‘size_t’ [-Wformat=]

The most interesting part of the `strlenArray.c` code is this:

```
for (i=0; str[i] != '\0'; i++)
    length++;
```

The for loop visits all string characters in order until it reaches the terminating character of the string. It then exits the for loop and returns the length of the string. The most exciting part of `strlenPointer.c` is the following code:

```
while ( *str )
{
    length++;
    str++;
}
```

The `str++` code is the way to visit all string characters using pointer arithmetic. Similarly, the `*str` code returns the current character of the string. The `while()` condition becomes false when it reaches the terminating character. Another way to write `myStrlen()` using pointers is:

```
size_t anotherStrlen(char * str)
{
    const char *sc;
    for (sc = str; *sc != '\0' ; ++sc)
        ;
    return (sc - str);
}
```

This method is a little obscure – avoid it unless you know exactly what you are doing. It subtracts two pointers and returns the result of the subtraction; nasty bugs can happen this way, despite the fact that this particular version is not only correct but also the most compact of all possible implementations. Find the source code of this article at github.com/mactsouk and FileSilo.co.uk. ■

INTERVIEW MARK SURMAN

Encryption is key

Do you really know how safe you are online? Mark Surman discusses Mozilla's launch of one of the web's biggest encryption education campaigns

**Mark Surman**

is the executive director of the Mozilla Foundation, pioneering work to help keep the web open and free for people all over the world. He is also a loud proponent of universal web literacy.



Staying encrypted

There's a myriad of ways that users can encrypt their web life, and most don't require much work or outlay on the user end. VPNs have always been touted as the easiest way to stay safe online – not only do they add an extra layer of security to your network, it's also an invaluable tool for protecting your privacy. Also, consider the browser you use; while Firefox is undoubtedly an encryption-friendly choice to make, the likes of Tor also come highly recommended. Through its additional software, Tor is capable of relaying your web traffic through several other computers before arriving back at its destination.

**Could you tell us the key principles of the encryption campaign that Mozilla is launching?**

Ultimately, we set out to create an education campaign. We want to make public understanding of encryption more mainstream and easily recognisable for people all over the world. When you then get more people involved, it's easier to get people to understand and get involved with the concepts around encryption. It's a term that many people would have heard of, but understanding what it means for them is a whole other concept.

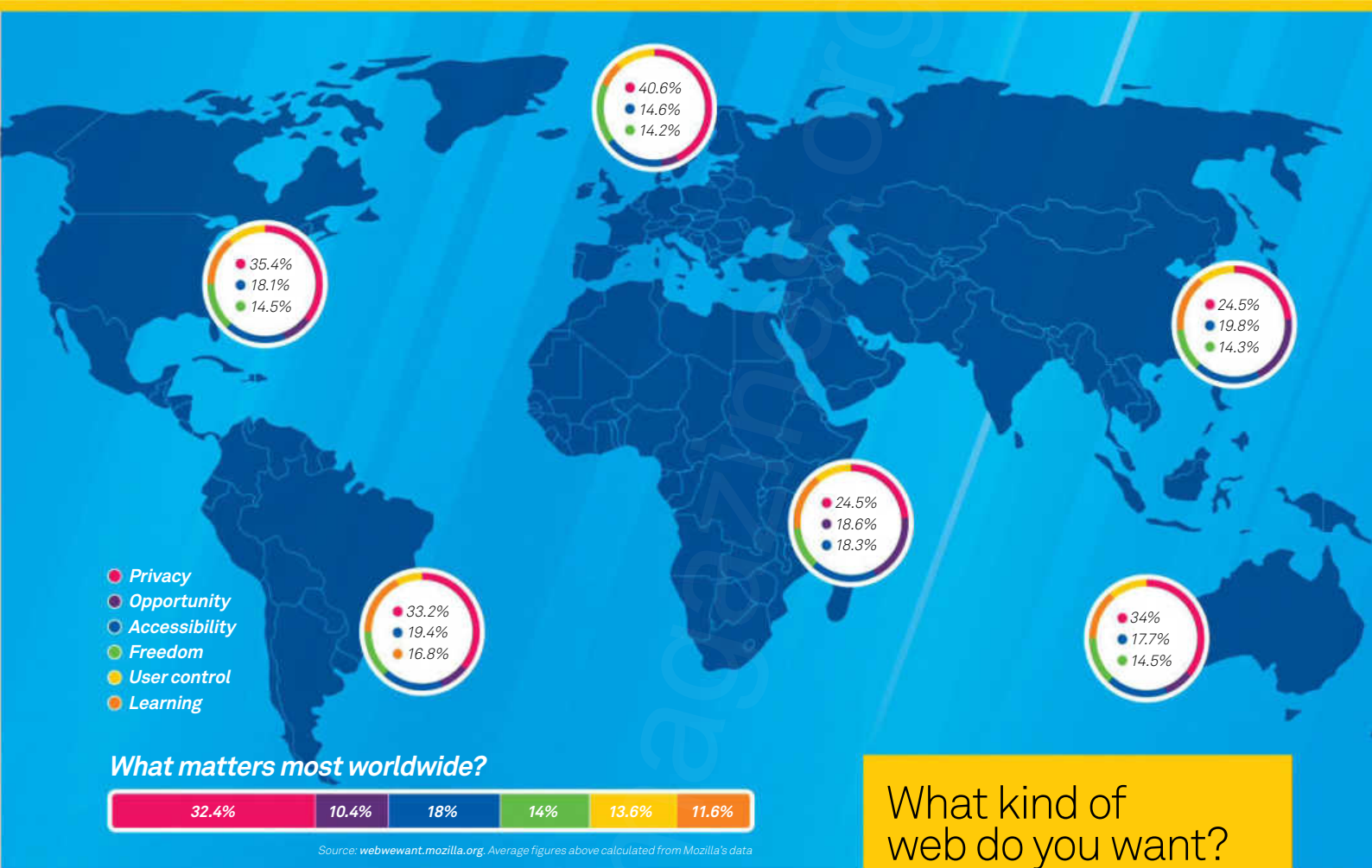
Mozilla wholeheartedly believes encryption is key to a healthy internet. It's a crucial part of both online security and privacy at the most basic of levels and it allows us to more safely bank, shop and communicate online. Encryption plays a big part of our everyday lives, no matter how you use the internet, and is something more people need to carefully think about.

At the core of the campaign is a set of videos that cover these topics in detail. Our first video spotlights what a world without online privacy might look like. While it may look goofy it actually makes a serious point, centred on how controlling our personal information

online is absolutely essential. Things like identity fraud are becoming more commonplace each and every day, so it's never been as important as today. The second video casts encryption as an animated character, and explains encryption's role in the lives of everyday internet users. Again, it's tailored for those who may not fully understand encryption but want to know more. There are plans in place for further videos, and we advise readers to keep track of our website to watch these videos and learn more about the campaign: <https://advocacy.mozilla.org/encrypt>.

A lot has been said in the media recently about web encryption, but what was Mozilla's original motivation for putting this campaign together?

We see encryption increasingly coming under threat from different people and programmes all over the world. Government agencies and law enforcement officials around the world are considering policies that could weaken encryption and their proposals could lead to problems for us down the line. We believe proposals to weaken encryption would seriously harm the security of internet users, especially those who are less aware



What kind of web do you want?

Mozilla's The Web We Want campaign is all about getting into the cold, hard facts about how people use the web and want they want from it. It's a great tool to help view which parts of the world care most about privacy, or learn about the most important issues that affect millions of internet users every day. But it goes even further than that; users can partner up different countries to compare usage statistics and see how different their web problems are. Mozilla's ambitions for a free, neutral and easily accessible internet has never been made clearer than here.



of the problems they could potentially face from this. Encryption plays a fundamental role in protecting financial, health, and communication data, among other things and it's imperative that we bring these issues to the forefront of every internet user.

The U.S. government's demand for Apple to circumvent their own security protections is a recent example of this. It is a massive overreach. To require Apple to do this would set a dangerous precedent that threatens consumer security going forward. The public can make a big difference on issues like this, which is a big part of the thinking behind the campaign. The open internet has been in danger before – SOPA/PIPA, threats to net neutrality – and public support was critical in turning the tide. The public has a bigger voice on issues like this than they may initially think, and it's important that they are heard.

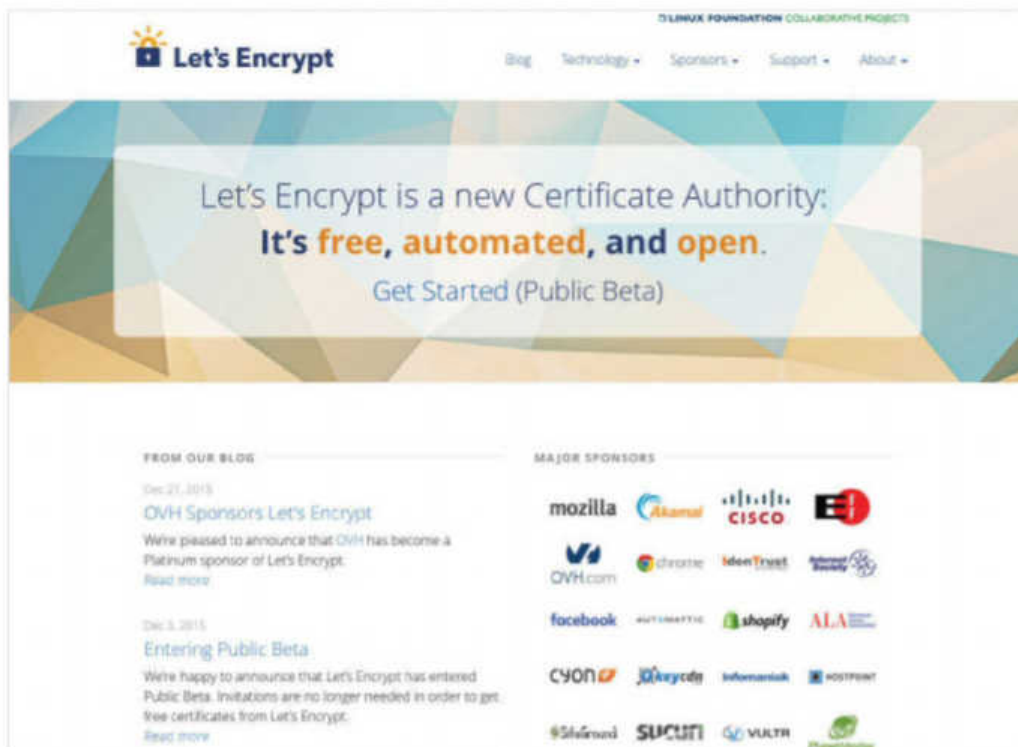
You've already mentioned the video creation side of things – what else will you be doing with the project going forward?

We'll continue to release videos, blogs and more through to the end of March. But the campaign's

“The open internet has been in danger before – SOPA/PIPA, threats to net neutrality – and public support was critical in turning the tide”

What is Let's Encrypt?

The Let's Encrypt project is a certificate authority aimed at making encrypted connections to the web as ubiquitous as possible, ultimately lowering the complexity of setting up TLS encryption. The primary benefit of the Let's Encrypt authority is that it provides the necessary framework for companies to automatically obtain a browser-trusted certificate, but without any human intervention. There's no configuration editing or validation emails, or any prospect of expired certificates taking your site offline, thanks to auto-renewal. We should also mention it's completely free – a giant step in a new direction for website security. As far as Linux Foundation collaborative projects go, this is one of the best ones that will be of benefit to millions of site owners the world over. See letsencrypt.org for more info.



conclusion won't mark the end of our support for encryption. Mozilla will always advocate for strong encryption and a secure internet. We've been pioneering that with some of the work we've done in Firefox, and will always look to work on smaller projects with other people down the line.

What do you hope to accomplish with your encryption campaign? A safer internet for regular users, perhaps?

Our aim is to build a firm public understanding of encryption in a way that makes it more accessible than what's currently out there. For the most part, a lot of what's out there currently is either not easily accessible, or not easy to understand. Once the new material is in place, we can begin to have a public dialogue about why weakening encryption is a bad idea. And, if necessary, regular internet users will be prepared to stand up for encryption and voice support. It's essentially playing the long game, but if it means advocating change in the future, then we want to play a big part in it.

How easy is it for users to get directly involved with the project?

It's as easy as sharing our campaign videos with friends and family through social media channels. Our goal is to start meaningful conversations around the kitchen table, in comment sections, and elsewhere. Users are also able to sign up via the Mozilla site for access

“Our goal is to start meaningful conversations around the kitchen table, in comment sections, and elsewhere”

to further resources and other topics that cater for protecting the web.

You're also noted as one of the main sponsors for the Let's Encrypt certification authority. Could you tell us more about that and Mozilla's role within it?

Let's Encrypt (<https://letsencrypt.org>) is a free, automated web certificate authority dedicated to making it easy for anyone to securely communicate over the internet. It's been developed in collaboration with the Electronic Frontier Foundation, Cisco, Akamai and many other technology organisations. Let's Encrypt is an example of how Mozilla uses technology to make sure we're all more secure on the internet. There's currently an open beta for interested users to check out and install on their machine, which is highly recommended you try if you've ever gone through the trouble of setting up a secure website.

Do you see any crossover between the Let's Encrypt certification authority and your own privacy awareness campaign?

There is definitely an affinity. The campaign is educational and consumer-focused in nature, and Let's Encrypt is a technical undertaking – but they share the same goal. Both projects are working toward creating a more secure internet, which is something that most web-based companies can fully get behind and inspire the millions of internet users around the world to also get behind. ■

EXPLORE THE TECH INSIDE

www.gadgetdaily.xyz



Gadget

Available
from all good
newsagents and
supermarkets

ON SALE NOW

■ RACING DRONES ■ VIRTUAL REALITY GUIDE ■ POWER TOOLS



BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



facebook.com/gadgetmagdaily twitter.com/xgadget_magazine

OPINION

The kernel column

Jon Masters examines the latest Linux happenings as Linux 4.5 is released and the merge window opens for what will become 4.6

**Jon Masters**

is a Linux-kernel hacker who has been working on Linux for some 19 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 final release of Linux 4.5, noting that he “just couldn’t make up my mind whether I should do another rc8 [release candidate] or not, and kept just waffling about it. In the end, I obviously decided not to, but it could have gone either way.” As we reported in last month’s issue, the latest kernel includes fewer earth-shattering new features than are true of a typical release cycle (this just happens periodically during the year based upon when certain new features are developed and available) but it does have among the new features support for volatile memory ranges via `madvise()`, which we covered in detail in last issue, as well as an important copy offload optimisation provided by the new “`copy_file_range`” system call.

Using the newer system call, commands like “`cp`” (the Linux and Unix file copy shell utility used from the command line) will not have to extraneously perform a copy of data contained within a file into the userspace memory of the copy utility before writing it back out to a new file, saving an entirely pointless round-trip operation into and out of the kernel. Instead, Linux will perform the copy action as a single kernel operation. Still, thanks to years of other optimisations, file copies using the new system call become only marginally faster under many normal situations. However, when using network-based filesystems, such as NFS (Network File System), that also include server-side copy support (such as is now available in NFSv4.2, also added in the 4.5 release), the kernel saves transferring file data over the network twice as it is read from the old file into a new one. This can dramatically speed up file operations on large networks using NFS.

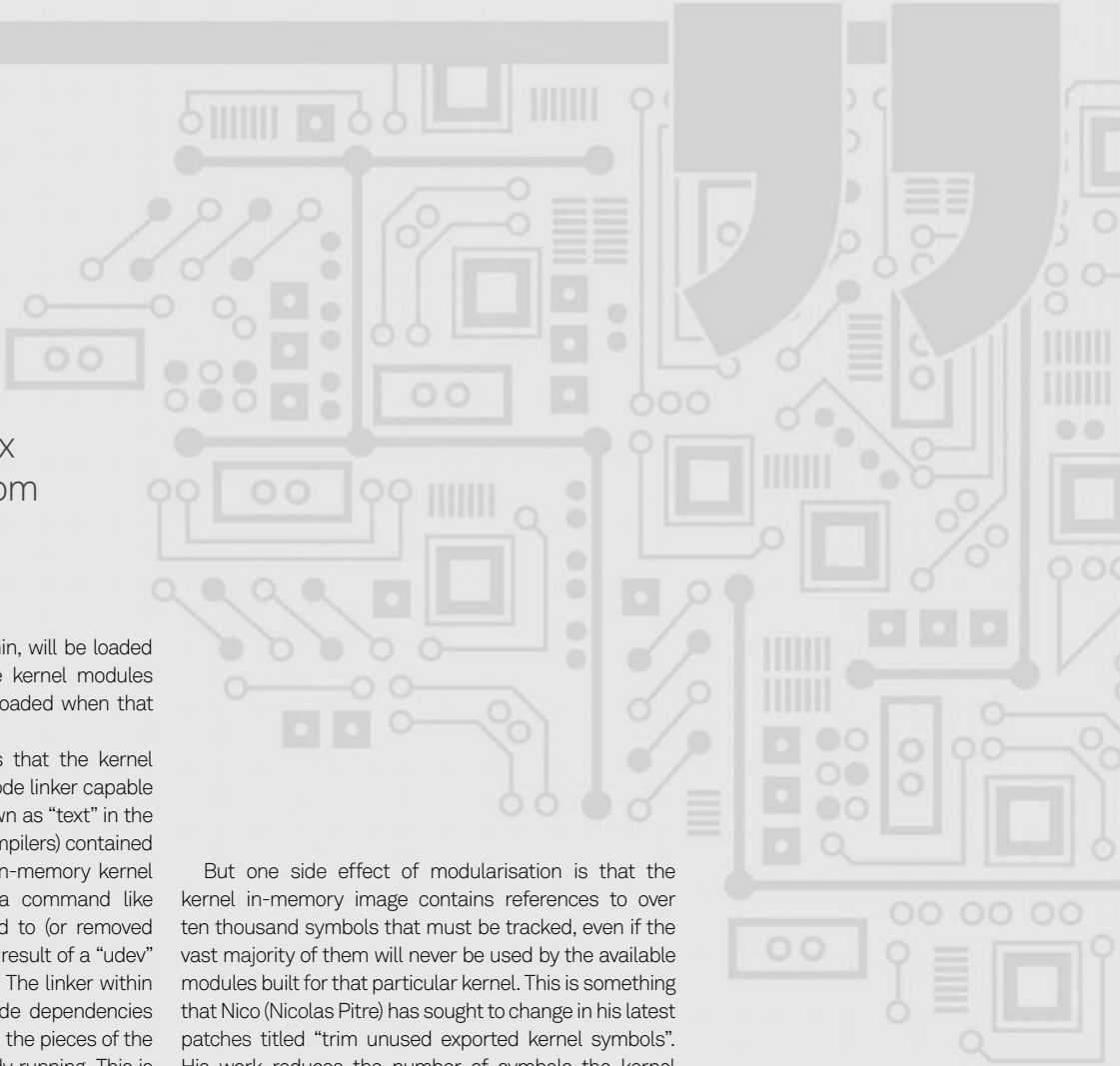
With the release of Linux 4.5 comes the opening of the “merge window” (period of time during which disruptive changes are allowed to be integrated) for what will be Linux 4.6. The pace of churn is quite small so far, given that many of the early features being merged are particularly low level in nature and thus more finicky to debug along the way. If things continue according to a normal kernel cycle, we should have a full summary of all of the features making their way into Linux 4.6 in time for next issue.

Trimming unused kernel symbols

In the early days of Linux, back when Linus and Andy Tanenbaum were having their now infamous arguments on microkernels vs monolithic kernels (which are worth digging up for their comedy gold in the online usenet archives – usenet being an older technology for group-based discussion that is also intentionally archived by Google Groups), Linux fell under the latter camp. Being monolithic meant that Linux built all of the code needed to support kernel operation into a single kernel binary image that was linked together as a tightly coupled unit. Contrast this with the concept espoused in microkernel design, in which a kernel is built from many parts that exist independently and communicate between one-another in much the same way that user applications communicate using interfaces under Linux.

Arguments around microkernel vs monolithic kernel design continue today in Computer Operating System classes and forums – as an example, Windows NT, and by extension modern Windows systems, is partially microkernel in its design, while the Darwin kernel used by OS X relies upon an underlying Mach microkernel upon which it is built. For the most part, though, the notion that Linux was misguided in choosing one particular path is mostly settled for the normal user base (Linux took off, after all). In part, this is because contemporary Linux isn’t strictly formed from a single inseparable, inviolable whole. Instead, such contemporary Linux systems make heavy use of modularisation – compiling kernel code into runtime dynamically-loadable modules – as a means to add and remove functionality while keeping a brighter line between the pieces.

Being modular in nature isn’t the same as the rigid separation imposed by a microkernel, but the end result of modularisation nonetheless gives Linux users the ability to consider the kernel as being formed from different components that communicate with one-another. It also allows Linux distributions to build kernel support for many more features than are available for (or needed by) any one particular hardware configuration – the kernel will load only the functionality that is required at runtime based upon hardware that is dynamically detected (or added and removed). If the user inserts a USB stick, then support for the device itself, as well as any



“Contemporary Linux isn’t strictly formed from a single inseparable, inviolable whole”

storage or other media contained within, will be loaded by the kernel automatically from the kernel modules contained on disk (and potentially unloaded when that USB stick is removed once again).

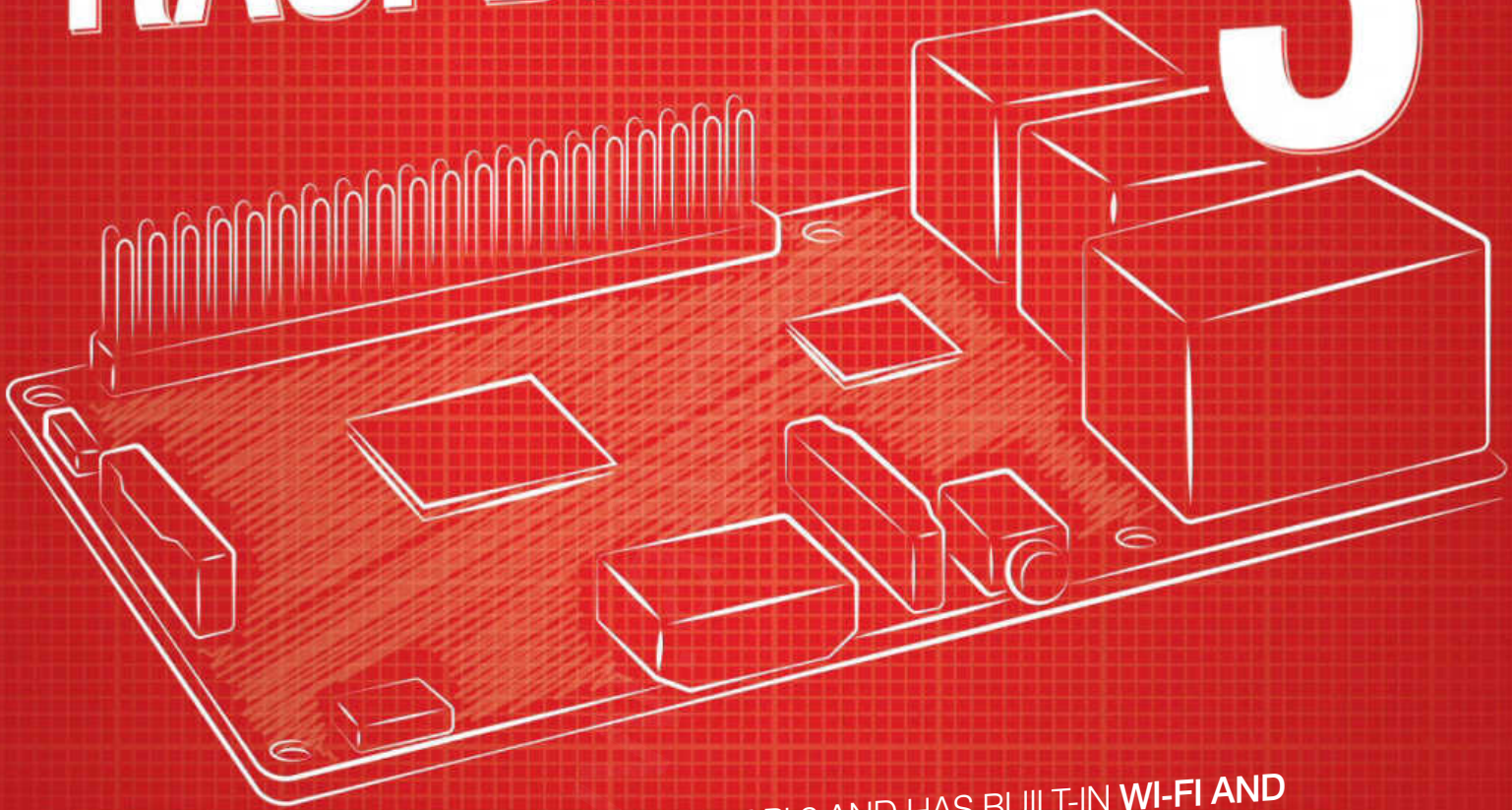
Supporting modularisation requires that the kernel include a powerful runtime dynamic code linker capable of adding and removing the code (known as “text” in the parlance of those who are engineer compilers) contained within the modules from the overall in-memory kernel image. This code is run whenever a command like “modprobe” causes code to be added to (or removed from) the running system, usually as a result of a “udev” detected device probe (hotplug) event. The linker within the kernel needs to resolve many code dependencies between a loadable kernel module and the pieces of the in-memory kernel code that are already running. This is known as symbol resolution, and dynamic linkers spend their lives resolving symbols (names of functions) into memory addresses that are patched into the loaded code before it begins to be executed.

Supporting dynamic runtime symbol resolution in the Linux kernel is hardly a trivial task. The Linux kernel includes more than ten thousand symbols (kernel functions) that have individually been annotated with the “EXPORT_SYMBOL” macro. When a kernel function such as “printk” (which causes messages to be logged to the kernel ringbuffer and output into the system log/journal) is exported with “EXPORT_SYMBOL(printk)”, the build-time compilation inserts a reference into the kernel image associating the location of “printk” with its memory location in such a way that loadable modules later using printk (which they nearly all do) can resolve their own use of the symbol back into its actual in-memory location to be patched up. The linker performs this patching up of memory as part of module load. Using symbols, then, allows the kernel to be very modular in nature.

But one side effect of modularisation is that the kernel in-memory image contains references to over ten thousand symbols that must be tracked, even if the vast majority of them will never be used by the available modules built for that particular kernel. This is something that Nico (Nicolas Pitre) has sought to change in his latest patches titled “trim unused exported kernel symbols”. His work reduces the number of symbols the kernel tracks down to only those actually referenced by the modules built for that particular kernel – taking the total down from over 10,000 to as few as a couple of dozen to a few hundred.

The net result of Nico’s work is a smaller, more efficient kernel binary that may be between 3 and 14% reduced in size. Importantly, when he combines his patch with something called “LTO” (compiler supported Link Time Optimisation – removing code that isn’t needed or inlining pieces that don’t need to be specifically exported symbols) of the kernel code during compilation, he can reduce the kernel binary image size by as much as 13.6%, which is close to the 13.9% number that comes with entirely disabling loadable kernel module support. This means that for embedded systems (such as small IoT devices), loadable module support effectively becomes free of overhead, adding much needed flexibility to the design of small resource constrained system designs that in the past might have been forced to disable modularisation entirely. ■

25 MAKER PROJECTS FOR RASPBERRY PI 3



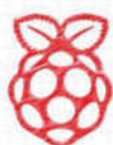
THE PI 3 IS **50% FASTER** THAN THE PI 2 AND HAS BUILT-IN **WI-FI AND BLUETOOTH**. WE SPOKE TO THE FOUNDATION'S DIRECTOR OF HARDWARE, **JAMES ADAMS**, TO LEARN MORE, PLUS WE'VE COME UP WITH **25 EXCITING NEW PROJECTS** USING THE PI 3'S WIRELESS TECH

TURN
TO **PAGE 26**
FOR OUR
REVIEW

WIFI &
BLUETOOTH

SAME FORM
FACTOR

4X ARM CORTEX-A53,
1.2GHZ



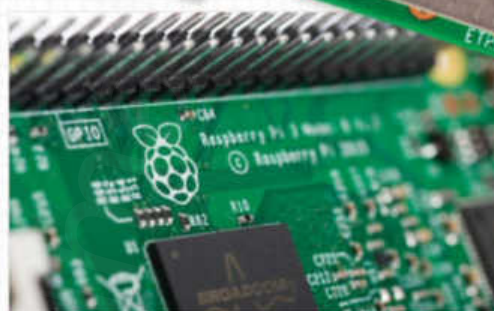
WHAT ARE THE BIG CHANGES?

A lot of the Raspberry Pi 3 is the same as the Raspberry Pi 2. Our strategy for the Pis from the B+ has been to try and keep the same form factor. The B+ was really the first Pi that I worked on when I came to Raspberry Pi in 2013, and the idea was that we've got this nice form factor and we want to keep it, so to build the Pi 3 I took the Pi 2 design and literally just added the extra stuff.

So, we've got the new processor courtesy of Broadcom – they've done a nice uplift. It's similar to the BCM2836; the Cortex A7 was taken out and the A53s were put in, and everything else has actually stayed the same so you've got backwards compatibility. The clock speed on the GPU has been uplifted slightly from 250 to 400 MHz, and to the PCB itself we've added the Wi-Fi and Bluetooth to the top-left corner. We had to move the LEDs, unfortunately, to the bottom-left corner of the board. There have been a few other little minor tweaks, but largely it is the same as the Pi 2 with Wi-Fi and Bluetooth. That doesn't mean it was a trivial thing to get going – actually, the Bluetooth and Wi-Fi was a big engineering challenge.

HOW MUCH HAS THE SPEED IMPROVED WITH THE NEW BCM2837?

We're saying about 50 per cent [from the Raspberry Pi 2]. In reality, you can usually see a bit more improvement than that – it really depends on what features of the processor things are taking advantage of. In everyday use, you'll get about 50 per cent faster, so about 33 per cent clock-to-clock speed increase on the core plus the uplift from 900 to 1.2 GHz. But then the NEON unit in



Left The new 1.2GHz BCM2837 contains four ARM Cortex-A53 cores

the A53 is, I think, double the width of the one in the A7, so if you have things that take advantage of ARM NEON then they'll go quite a lot faster. I think there are some other branch-prediction cache improvements in the core that, in things like web browsing, can make more of a difference – it really depends on the workload. The A53 is just a much newer and slightly fatter core.

You've got 64-bit, which we're not using at the moment, but we have had people come to us with preliminary Linux kernels running in 64-bit mode. I think Eben said publicly that we're just going to monitor that, see what we think once we see that running, whether we actually do our own 64-bit OS version, which would be completely separate from the current 32-bit, or if we keep 32-bit going forward, which is nicely backwards compatible with all previous Pis.

WHY WAS 64-BIT FUNCTIONALITY ADDED?

At the point we were putting a new core into the 2837, we were considering the options. All the new ARM cores basically have 64-bit support anyway – it's a nice



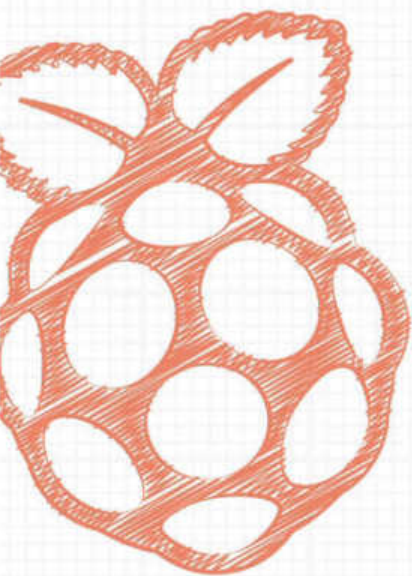
JAMES ADAMS

is the director of hardware at the Raspberry Pi Foundation, where he manages the current production hardware and develops new products. He first joined the team back in 2013 and since then has led the design of the Raspberry Pi Model B+, the 2B and the new 3B. It has been whispered that James is also a demon welder and brewer of beer.

NO FM

There's been speculation that the Pi 3 boasts an FM receiver, but this is sadly not the case. "The FM is effectively disabled," confirms James. "The Wi-Fi chip is a BGA device, so underneath you've got tiny bumps of solder and they have to solder onto the PCB. Now, on that chip they are pitched 0.4mm apart, which is actually designed for higher-tech mobile phone PCBs – higher-tech than the board that the Pi uses. One of the reasons we got the cost down is because we use low-tech PCB design, so actually, through some clever tricks, we've managed to use that higher-tech part on the lower-tech board, but at the expense of FM – too many signals to route out, and you have to resort to a higher-tech PCB."

Right James is also a part of the FiveNinjas team that made a Compute Module-based media player



feature for the future – but we're very, very committed to keeping form factors and software backwards-compatible, at least as far as possible. So currently there are no plans to do a 64-bit version of Raspbian. It will require two completely separate OSs that we'll then have to manage. We'd have to recompile everything in userland to be 64-bit, so it's an awful lot of work, and then you've got the support burden as well.

Technically, it's possible, and we'll see what performance gains 64-bit Linux has, and then we'll take a view some time in the future as to whether it's worth it or not. Obviously, 64-bit ARMv8 is the way that ARM cores are going, and all future ARM cores will support it – so it's a thing, but we don't have an official position. We're going to see what happens; I think it's a fine way to be.

A FEW PEOPLE, LIKE OUR REVIEWER GARETH HALFACREE, HAVE NOTICED THAT THE PI 3 RUNS HOTTER THAN ITS PREDECESSORS – SHOULD PEOPLE RESTRICT ITS UPTIME OR IS IT STILL FINE FOR EXTENDED USE?

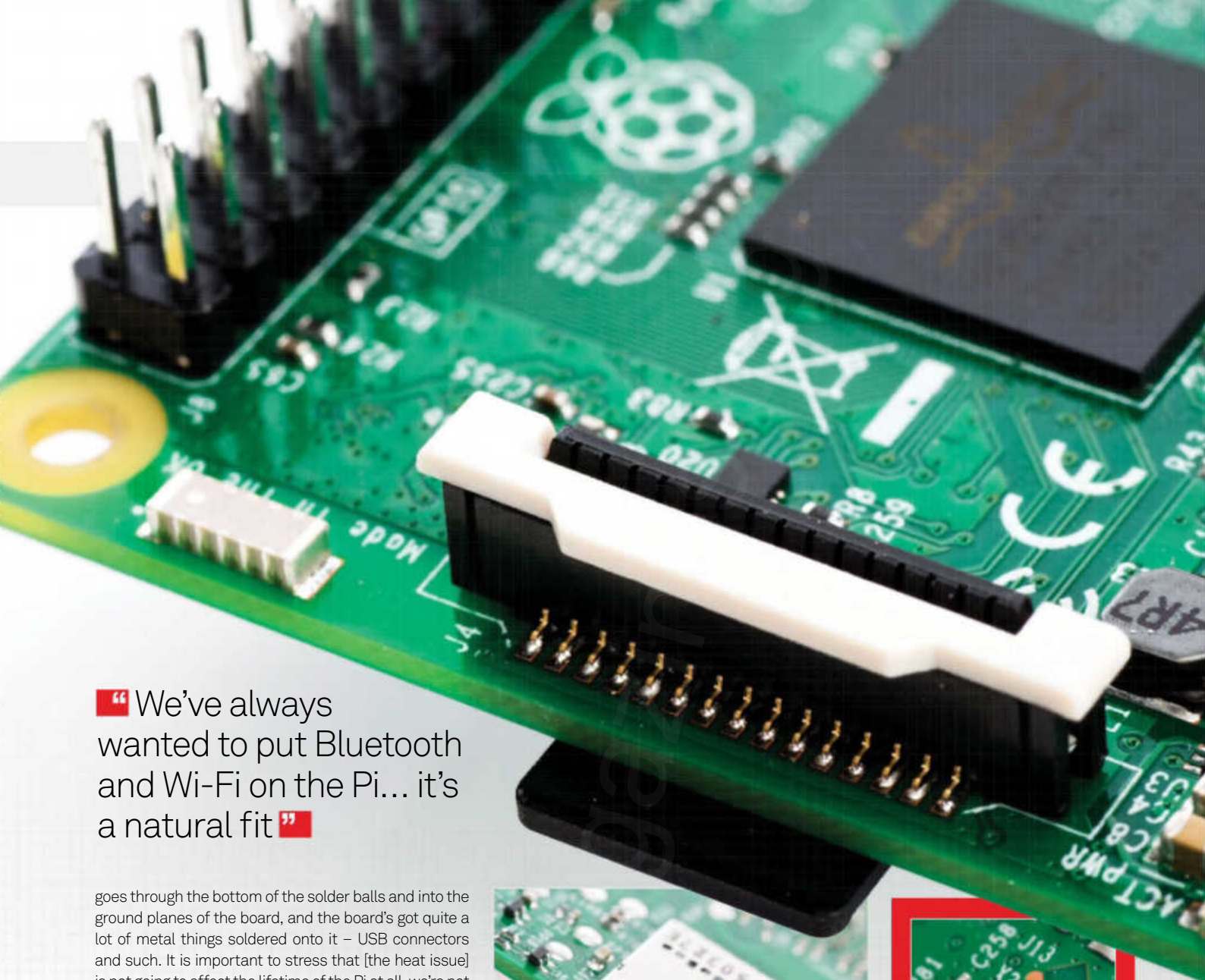
The heating issue is something we can reproduce in a special case. One thing I will say is that the Raspberry Pi 2 is similar – it just takes a little bit longer to heat up. In terms of what's actually changed, the frequency has gone up a bit and we've got a slightly bigger ARM core, so that core does generate more heat. Now, the Pi 2 core still generated quite a lot of heat compared to the Pi 1, if you run it flat-out in the kind of use cases that Gareth

has been playing with. But will it kill your Pi? Absolutely not. The 2837 SoC is qualified for 125°C temperature running for, I think, ten years of life – that's where they cut it off and say 'That's fine' – so we're not really worried about the hardware falling over.

The Pi will throttle the CPU speed when it reaches a hot temperature – this is set to 85°C in the firmware. We're still talking to Gareth to see why he seems to get his Pi up to a slightly hotter temperature. There is a bit of a temperature gradient across the chip – the on-die temperature sensor is in one corner of the chip – but in any case, we're not worried about it and we may update our temperature algorithms to be more accurate, given Gareth's data.

Having said that, though, these chips can get hot – especially when pushed hard, and they will throttle; they'll drop their frequency and voltage. So a Pi 2 or 3 would benefit from a heat sink to reduce the amount of throttling if you are using it in a very heavy workload continuously, whereas the Pi 1, you could run it flat-out all day and it would barely get warm. There are now four significantly fatter cores running at a much faster rate, so we're just bumping up against physics.

Your mobile phones work in the same way – they do this sprint performance, where they'll ramp the core right up to as fast as possible, your phone will get warm, and as soon as it's getting to the point where it's too hot, they'll turn the frequency and voltage down. But for the standard 'bursty' workloads, like web browsing, the chip on a Pi cools down really quickly if you stop doing a lot of work, so the board itself is quite a good heatsink. Heat



“We’ve always wanted to put Bluetooth and Wi-Fi on the Pi... it’s a natural fit”

goes through the bottom of the solder balls and into the ground planes of the board, and the board’s got quite a lot of metal things soldered onto it – USB connectors and such. It is important to stress that [the heat issue] is not going to affect the lifetime of the Pi at all, we’re not really worried about that.

WHEN WE ASKED ABOUT WI-FI AND BLUETOOTH LAST YEAR, THE FOUNDATION SAID THAT IT WOULD, ESSENTIALLY, BE TOO EXPENSIVE TO INCLUDE THESE COMPONENTS. WHAT’S CHANGED?

There are a few factors. We’ve always wanted to put Bluetooth and Wi-Fi on the Pi because it’s a natural fit – most people who buy a laptop now will have Wi-Fi; wired Ethernet is sometimes not even supported on laptops, if you buy from Apple. So it’s an obvious fit. Now, the challenge with that is obviously the price. We worked very hard on the Pi 2 building materials to work with suppliers and get some cost out of it, so we could afford to put the device on. We’ve managed to make a space in the costing to fit the Wi-Fi, and Broadcom’s Wi-Fi solution is really great because it’s quite low-cost, doesn’t use a lot of external components.

The other side of it is actually that the radio design, development and conformance testing is a very big engineering task. As soon as your product starts radiating, you run into a whole big pile of conformance issues – essentially, every country wants you to do testing to be within their radio spectrum rules, and it just puts a lot more burden on the tests. So we had to work



Left Among the minor changes is a return to the old slide-lock microSD card slot – no more push-to-eject

very closely with Broadcom to both set the chip into all the required test modes for the test houses, which was an awful lot of effort. And also just general RF design is harder than standard design, because if you have things like electrical crosstalk or your power supply is next to something noisy, it can really affect the radio performance; they’re much, much more sensitive than general digital designs.

It was a little bit more of a challenge than usual – all in all, just quite a lot of work to put RF on something! Most of the time was spent on that – despite the fact that we’ve changed very little on the Pi, there is probably as much engineering effort going from Pi 2 to Pi 3 as there was from B+ to Pi 2. The nice thing is that now we’ve done it, we’ve gone quite a long way up the learning curve on how to do it again – as Raspberry Pi, we now have those engineering skills.



U.FL ANTENNA

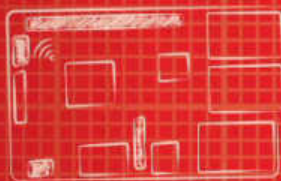
The conformance testing involved radiated and conducted modes. With the latter, “rather than pumping it into the aerial and spraying the RF into the air,” James explains, “you plug a cable into the tester. For that, you need to have some kind of connector on it. There’s a tiny shorting resistor you connect to the U.FL connector, and you can then connect an antenna or, in our case, all the test equipment we needed. I would say that if you want to use the U.FL antenna, don’t – the FCC don’t like it. But, *theoretically*, you could scrape the soldering mask off, solder a U.FL on, and connect an antenna.”

25 MAKER PROJECTS

TAKE ADVANTAGE OF YOUR ON-BOARD WI-FI AND BLUETOOTH WITH THESE NEW AND UPGRADED PROJECT IDEAS

1

WILDLIFE CAMERA



While you can pick up a kit for this, like the one from Naturebytes (bit.ly/1PdUNue), it's easy to put your own wildlife camera together. Make sure that you have a waterproof case to protect your Pi, the portable power supply and your other electrical components against any damage from rainwater; there are 3D models

(STL files) available, which you can easily tweak to fit the Pi 3 using a simple program like 123D Design. Then you just need to use waterproof sealing on any joints and on the hole through which the camera module points – check out this guide: bit.ly/1mgUA2X.

You can set your Pi up to shoot during particular times by triggering a Python script with a cron job – just add a real-time clock module via your GPIO header so the Pi knows what time it is while offline. If your Pi is within reach of your Wi-Fi, it can automatically upload or tweet photos for you. If you would prefer your camera to only shoot when there is an animal to capture, set up a PIR (passive infrared sensor) to monitor for changes to the infrared picture it is seeing, which would indicate a moving source of (body) heat.

With the Pi 3, you can hide your camera enclosure away in some really tricky hiding places (buried in bushes, tucked up inside trees...) and still have easy access to it via the on-board Wi-Fi and SSH if you need to make any changes to your scripts.

Parts

- Waterproof case with a camera mount
- Portable power pack
- Camera module
- PIR sensor
- Real-time clock



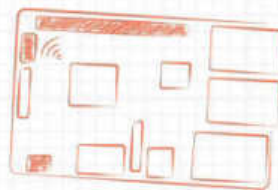
BORROW A SCREEN

Connect to a PC via Ethernet and use Remote Desktop. You still have internet access via the Wi-Fi.



2

WEATHER STATION



Parts

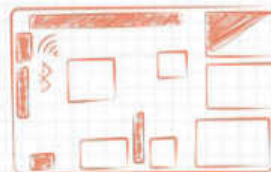
- DHT22 sensor
- BMP085 sensor
- TGS 2600
- UV-01 sensor
- LDR (light dependent resistor)
- ADC (analog-to-digital converter)
- Wind vane
- Anemometer

While a waterproof case is also a good idea here, some of the sensors would benefit from being outside the case, like the temperature sensor – there are waterproofed variants available for some of them. You could always use a mix of sensors, so you can take an average of the readings and also have some built-in redundancy.

The idea here is to connect up lots of sensors via the GPIOs in order to get a reading of the weather state in your area. With the Wi-Fi and portable power pack, you can have the weather station in your garden connected to your home router (or a Wi-Fi repeater!) for an internet connection, and you can post all of the data onto an easily-accessed WordPress site, hosted on the Pi.

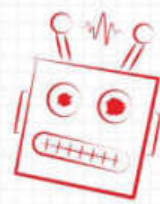
3

ROBOT



Parts

- Wi-Fi dongle
- Bluetooth speakers
- Bluetooth headset



Now, we've all been building Raspberry Pi robots for a while now – but with on-board Bluetooth and Wi-Fi? For a start, you don't need a Wi-Fi dongle in order to access your robot and make any changes to the control script on the fly, meaning that you can reduce the footprint of your design. And if you *do* use a dongle then you could even drive your robot around to use as a mobile Wi-Fi signal repeater! By setting up the BlueZ stack, you could also use your robot to control Bluetooth LE devices – perhaps mounting a pair of speakers on the back of your bot – as well as use Bluetooth devices such as the Wiimote to control movement (see bit.ly/1RRh4Qu). Get yourself a Bluetooth headset and you could even set up a voice-controlled application using something like Jasper (<http://jasperproject.github.io>), which listens out for your spoken commands.

GO WIRELESS

Invest in a Bluetooth keyboard/mouse combo and you'll free up all four of your USB ports

4

3D PRINTER

You can control a 3D printer using the Pi 3, and the extra horsepower means that you can handle the slicing on the Pi as well. First of all, you'll need to either set up a bought kit or look into the RepRap project – RepRaps are 3D-printable 3D printers, so you just need to find a friend with a machine or a high street 3D printer in order to run off the parts, then pick up the non-printable components (start here: http://reprap.org/wiki/RepRapPro_Huxley).

With your 3D printer set up, install the OctoPi image onto your SD card and then use the included software, such as OctoPrint and CuraEngine, to handle your

G-Code and slicing, driving it all via Wi-Fi. You could even set up the camera module inside the 3D printer and record time-lapse footage of your prints, though you'll want a powered external hard drive as well to store the video files.



Parts

- 3D printer
- Camera module
- Powered external hard drive

WI-FI HOTSPOT

5

Set up your Pi as a wireless access point for other devices, which assigns IP addresses. Connect it to the internet via ethernet, and it will act as a bridge for the devices connecting via its Wi-Fi module. If you're trying to get a connection down into your garden, use a portable power pack and plug in a USB Wi-Fi dongle. You then use the on-board Wi-Fi to connect to your home router and the plug-in Wi-Fi to broadcast into your shed.



Parts

- Portable power pack
- Wi-Fi dongle

SUPER SERVER

6



The Raspberry Pi 3 can keep up with huge amounts of page requests – the Pi Foundation ran an experiment during the Pi 3 launch to see if it could keep up with the huge number of visitors (<http://bit.ly/22mCjRR>), and over 12 hours it served 1.5 million requests. So, even if your website is *huge*, you can serve it from your Pi, and now it's got Wi-Fi you can configure it remotely while it's tucked away in a drawer somewhere.

PRACTICAL GADGETS

8 PIBOOK

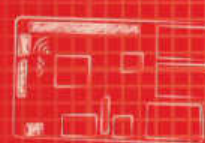
Fit a display into a 3D-printed laptop case, along with a keyboard and trackpad. Use tiny Bluetooth speakers for audio and connect to the internet via Wi-Fi. Switch to a portable power pack if you want to go mobile. Run something like Ubuntu MATE if you want to complete the look.



9 PIPAD



Mount your Pi on the back of the official touchscreen display and wire it all up. Use a regular power supply, and if you want to be untethered then switch to a portable



power pack. Be sure to pair it with a Bluetooth keyboard so you can get to the command line.

10 SMART TV

Install Kodi, then mount your Pi onto the back of your TV. Connect your TV to the Pi via HDMI, using adaptors if your TV is DVI/SVGA. Plug in powered external storage to add videos, and set yourself up with some streaming websites. Configure a Bluetooth remote or grab the Kodi app.



11 SPEAKER



Connect your speakers to the Pi via a Class D amplifier. Fit the setup inside a case, leaving a space for the power cable or including a portable power pack. Set up a Logitech Media Server on your Pi and control playback using the Squeezer app. Owners of a Bluetooth speaker can use BlueZ.



7

MEGA MEDIA CENTRE

Get your monitor, speakers, keyboard and mouse plugged in, then install OSMC – this is your main home theatre PC. Configure a Bluetooth remote control, or a remote control app on your smartphone or tablet. Connect a powered external hard drive to your Raspberry Pi, loaded up with all your favourite music, movies and TV shows. Set up your OSMC media centre by adding the content stored on your external hard drive, and add new files from USB flash drives as and when you plug them in. Set your Pi up

as a media server for your mobile devices. Finally, browse through OSMC's App Store for things like a torrent client and TV tuner.



Parts

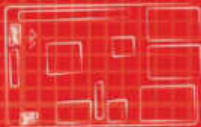
- Home theatre peripherals
- Bluetooth remote
- Powered external hard drive



SMART HOME TECHNOLOGY

12 AUTO-LIGHTS

Many Bluetooth LE light bulbs can be reverse-engineered to work with BlueZ – find out which library to download to work with it at bit.ly/1UjxGq8. You could also use Bluetooth home automation switches – plug your non-smart lights into those and then control the switch itself.



13 PLANT MONITOR

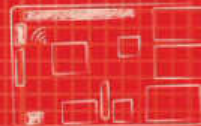
Use an SHT10 soil monitor to sense the temperature and moisture of the earth around the roots, then feed that data to a web server on your Pi that you can check via your phone.

Add a portable power pack and a waterproof enclosure for the Pi, and you can leave it in the garden.



14 TEMPERATURE CONTROL

Wire sensors like the DHT22 to your Pi, stick everything inside a case, then move the unit around the house to get readings. Upload all the data to a database on your Pi's web server to access it from a web page, and find out where the hot and cold spots are in your home.



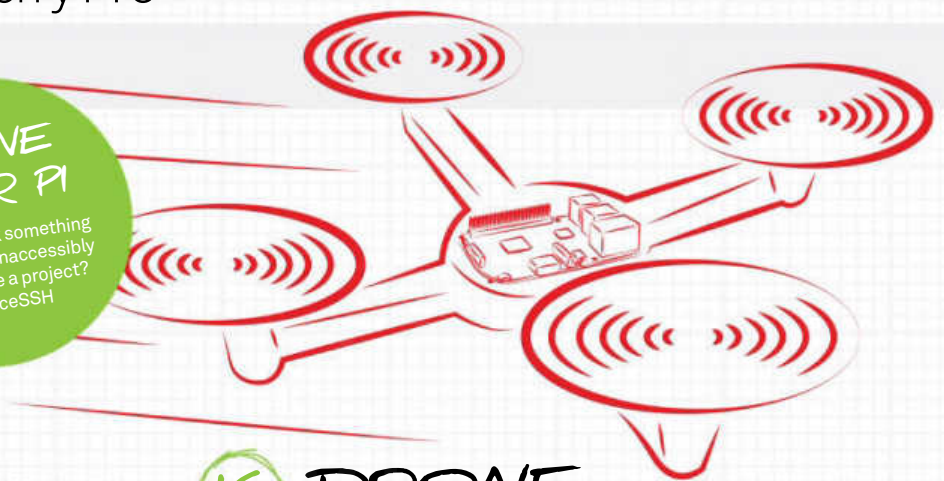
15 RC SOCKETS

WeMo switches plug straight into your wall socket. You then plug your non-smart appliance into that socket, and can remotely switch it on and off by sending Bluetooth commands from your Pi over the BlueZ stack. Combine this with cronjobs, or a mic and PocketSphinx...

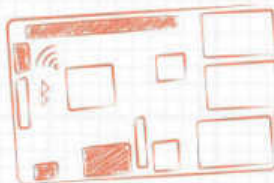


PHONE YOUR PI

Need to check something but your Pi is inaccessibly buried inside a project? Use JuiceSSH



16 DRONE



Parts

- Drone kit
- PXFmini autopilot shield

There are some great resources if you want to build your own drone – places like diydrones.com and instructables.com – including all the motors you'll need and 3D models for a chassis. Or you could instead purchase a complete drone kit. To actually power and drive the drone, you can use your Pi 3 with the new PXFmini autopilot shield. It's an excellent device designed for the Pi Zero

but is perfectly compatible with the Raspberry Pi 3, and its extra CPU power gives it a significant performance boost over a Pi Zero or Pi 2 drone. The PXFmini includes a gyro and compass, accelerometer, magnetometer, pressure and temperature sensor, plus an ADC – a very handy, compact bit of kit! And with the Pi 3's on-board Wi-Fi and Bluetooth, you won't need any extra modules in order to control it while it's in the air.

17 ON-LOCATION RECORDING STUDIO

Parts

- Cirrus Logic Audio Card
- Microphones
- Instruments



Add-ons like the early Wolfson Audio Card and the Pi-3 compatible Cirrus Logic Audio Card enable you to handle audio far better on your Pi. With an audio card, you can take advantage of tiny on-board mics or plug in your own to capture audio. There are also jacks for stereo line input, into which you can plug your instruments to capture high quality audio directly. You can also plug in external amplifiers or powered speakers. The power of the Pi 3 means that you'll be able to handle the recording without slowdown, and if you fit a display as well (or use a VNC app) then you can use Audacity to edit your audio on-location.

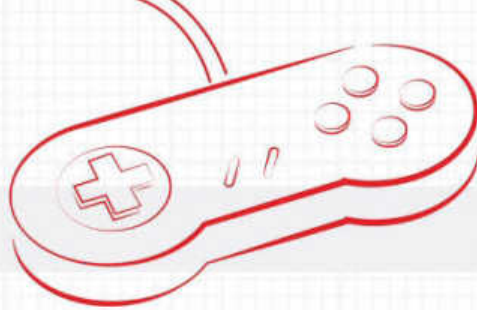
18 LIVE AUDIO PERFORMANCE KIT

Set up your speakers using the Class D amplifier and you've got your audio output sorted. Connect your instrument via the USB sound card, and then set up the Guitarix: this is a virtual guitar amplifier that can take your raw input, add an effect, then deliver a processed stereo signal out to your speakers. If you pre-set a few different effects, you can set them up to be triggered by your GPIO-connected push-buttons, or a button panel add-on. Again, with the souped-up Pi 3, you can now handle this level of audio processing comfortably. You could even have the Pi automatically upload recorded tracks to your Dropbox via Wi-Fi.

Parts

- USB sound card
- Button input
- Class D amplifier
- Speakers





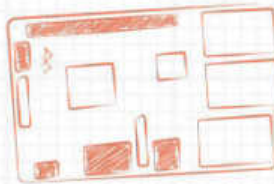
19 ARCADE CABINET

Parts

- Cabinet (bought/made)
- Monitor
- Speakers
- Joystick & Buttons
- Gamepads

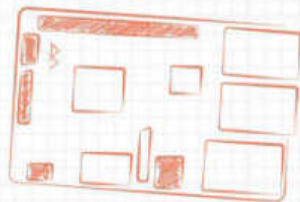
Grab an HDMI monitor (or DVI with an adaptor) and build it into a chassis for the cabinet, along with speakers. Fit the joystick and buttons, then wire them up to the GPIO pins. Install RetroPie on the Pi, add the ROMs that you own, configure the joystick and buttons, then configure any Bluetooth or USB gamepads you have for additional players. Fit the Pi into the cabinet – if you

need to adjust your setup, you can either use SSH, a Bluetooth keyboard/mouse, or leave access to two USB ports for a keyboard and mouse.



20 ARCADE TOUCH TABLE

Wire up the official touchscreen via the GPIOs and configure the display for touch input. Build it into the surface of a table. Add capacitive touch sensors in a D-pad plus buttons layout and install speakers. Now follow the method used by Frederick Vandenbosch for the surface (see bit.ly/1lwEMJ8) – lay down a sheet of clear Plexi over your capacitive touch sensors, add a couple of layers of paper to obscure the components,



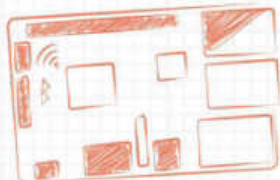
then another layer of Plexi. The capacitive touch sensors should be sensitive enough for you to trigger them by placing your hands over the surface, and you can even shine LEDs through the paper. Now set up RetroPie.

Parts

- Coffee table (bought/made)
- Official 7" touchscreen display
- Speakers
- 6-8 Capacitive touch sensors (Pi Desk ones)

21

CAR ENTERTAINMENT



Parts

- Touchscreen display
- Back seat displays
- Speakers
- Bluetooth keyboard/mouse combo

Set up a Kodi media centre on your Raspberry Pi and add a touchscreen display to make controlling it easy while driving. Install the unit into your car inside the radio slot, and add wired or wireless speakers, since you have your on-board Wi-Fi. Check out bit.ly/1jdL8rQ for a little inspiration. Put a Bluetooth keyboard somewhere handy and also grab any media-loaded flash drives you have. Install a Kodi remote app onto your smartphone, set it up and then use it to control the car stereo. You can also set up a portable wireless hotspot on your mobile, with your 3G connection, and then access your hotspot using the Pi's Wi-Fi module. For the finishing touch, run Navit with TTS output for a true sat-nav system.

ENTERTAINING ENTERPRISES

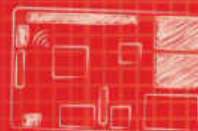
22 SMART TOYS

Find a medium-to-large toy and break into it so that you can hide the Pi. Now wire up your LEDs, camera module and mini loudspeaker, then seal the toy back up again. If you run a web server using the Pi's Wi-Fi then you can write a simple control interface to activate lights and sounds.



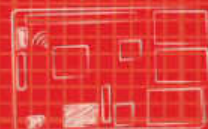
23 NERF SHUFFLE

Write a simple script to shuffle your music playlist on a given input, which in this case can be a shot from a Nerf gun. Wire up a medium vibration sensor so that it can detect a hit on your target, then pass that input to your script. Fire gun, hit target, change song. Give this a read: bit.ly/1RR9YLN.



24 WIRELESS PROJECTOR

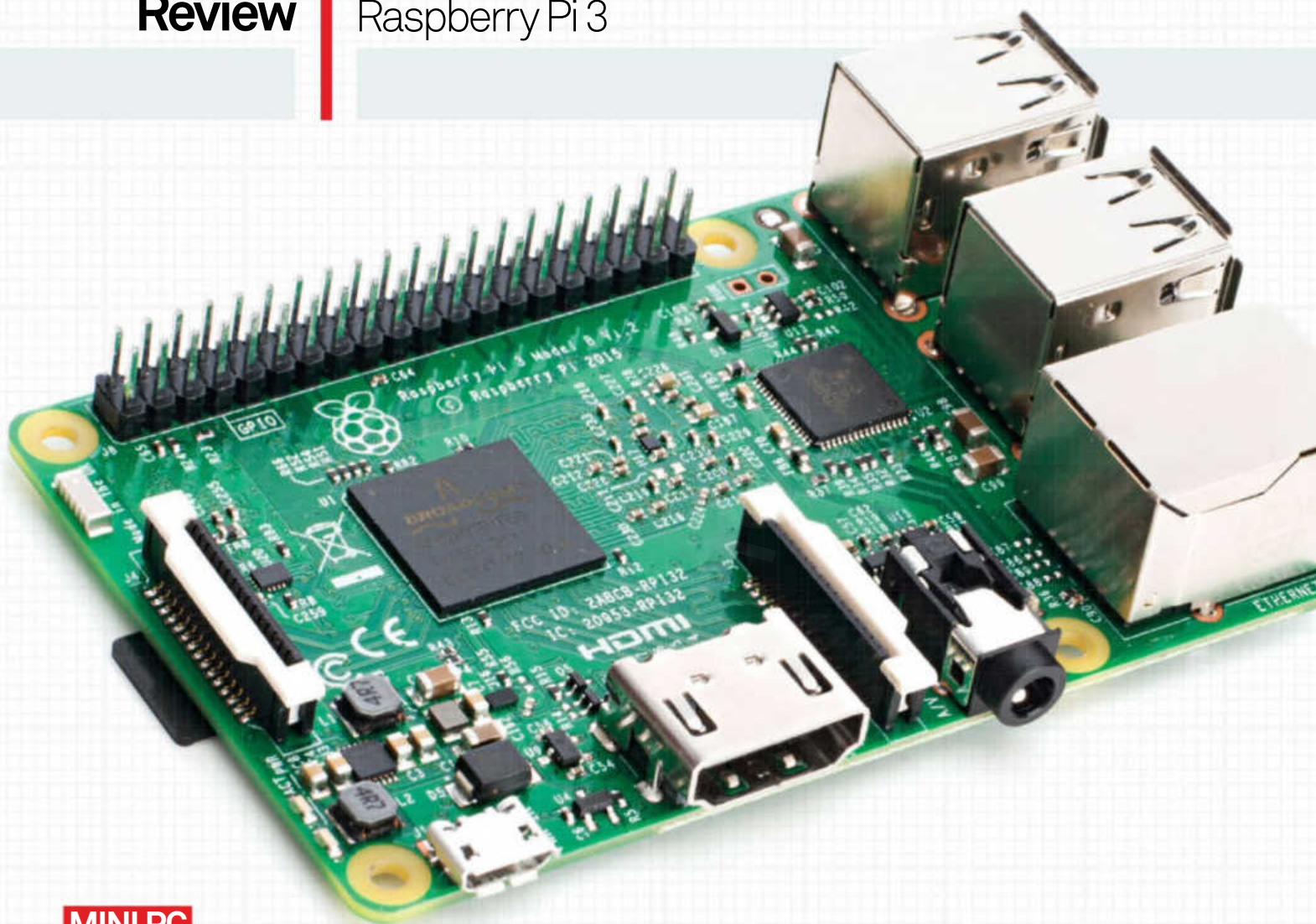
Get something like the Brookstone Pocket Projector and connect it to your Pi, stick the setup inside an enclosure and reposition a projected display on the fly. If the resolution isn't quite there, the Pi 3 can handle a smooth VNC connection to your computer over Wi-Fi.



25 MINECRAFT MAGIC WAND

Connect your Pi to the Wiimote via Bluetooth and then configure it with the CWiid Python module (bit.ly/1Wtzb2Z). With the Wiimote set up, you can now use the Wiimote buttons to trigger pre-made Python scripts to hack your Minecraft world.





MINI PC

Raspberry Pi 3 Model B

SoC

Broadcom BCM2837

CPU

1.2GHz, Quad-core ARM Cortex-A53

GPU

Broadcom VideoCore-IV

RAM

1GB LPDDR2-900

Storage

Micro-SD

GPIO

40-pin header, populated

Networking10/100 wired Ethernet, 2.54GHz
802.11bgn Wi-Fi**Bluetooth**

Bluetooth 4.1 Classic, Bluetooth LE

PortsHDMI, 3.5mm analogue audio-video
jack, 4x USB 2.0, Ethernet, CSI, DSI**Price**

£31.66 (cpc.farnell.com)

Released to coincide with the project's fourth birthday, is the Pi 3 everything the community has been hoping for?

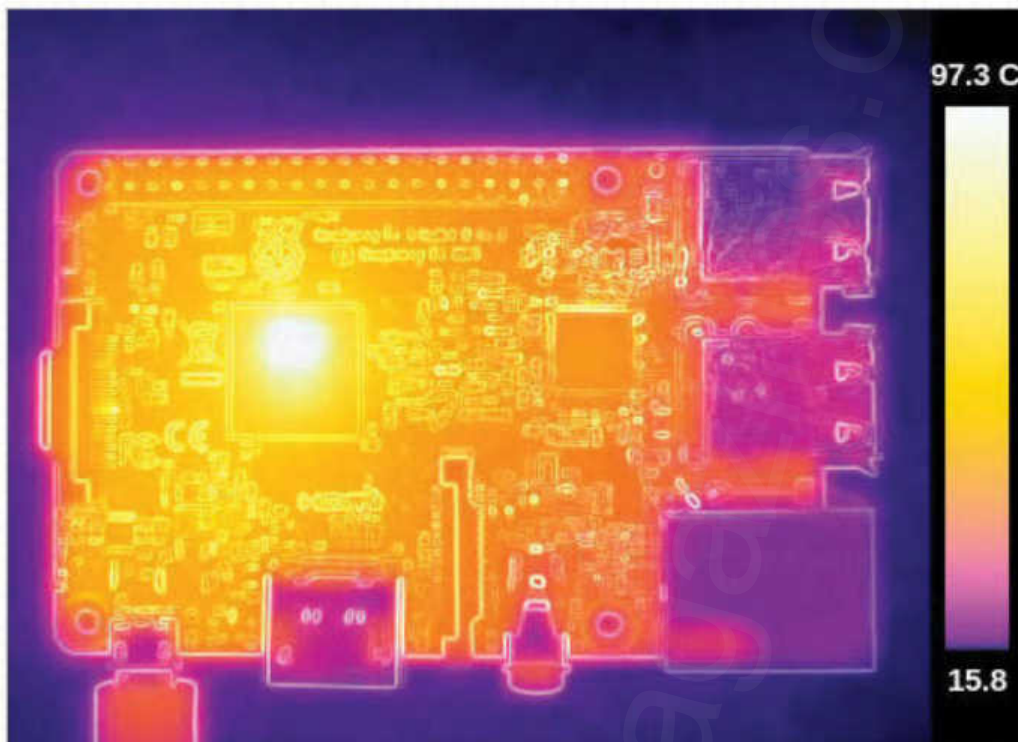
Launched on the day of the Raspberry Pi Foundation's fourth birthday – or first birthday, given that it was a leap day – the Raspberry Pi 3 is easy to dismiss at first glance. Its layout is near-identical to the Raspberry Pi 2 before it, but the presence of a small chip at the upper-left provides the first hint that things have changed.

The Pi 3 is the first Raspberry Pi to feature integrated wireless connectivity, and the Foundation hasn't done things by halves. The BCM43438 radio chip on the underside of the board provides 802.11bgn Wi-Fi – albeit in the 2.4GHz channels only – alongside both Bluetooth 4.1 Classic and Bluetooth Low Energy (BLE). All this is routed to the tiny chip antenna on the top of the board, which manages to impress with its performance for such a compact device.

The system-on-chip (SoC) at the heart of the Raspberry Pi 3 is new, too. Just as the Raspberry Pi 2 upgraded the original device's single-core ARMv6 BCM2835 to a more modern ARMv7 BCM2836, the Raspberry Pi 3 features a custom-made ARMv8 BCM2837 making the new Pi 3 the first in the family to boast 64-bit support.

Sadly, it's support that at present is going unused. The Raspberry Pi Foundation has indicated that it plans to maintain the current single release of Raspbian, its customised Debian distribution, compiled in the large for the old ARMv6 instruction set with a smattering of application-specific ARMv7 thrown in for good measure.

Despite this, the new Cortex-A53 cores offer a considerable performance boost over older models.



“If you’ve found the older Pi models restrictive, the new Raspberry Pi 3 could be just the device that you’re looking for”

The Raspberry Pi 3 can complete a multi-threaded SysBench CPU test in just 49 seconds to the Pi 2’s 76.28, and with the Model B+ taking 510.81 seconds to achieve the same it’s easy to see how far we’ve come. Even gaming gets a boost, despite the VideoCore-IV GPU remaining unchanged bar a slight clockspeed hike, with the *Quake II Arena* timedemo hitting 70.1 frames per second on a Pi 3, 57 on a Pi 2, and just 19.4 on the old Model B+. In short: if you’ve found the older Pi models restrictive, the new Raspberry Pi 3 could be just the device that you’re looking for.

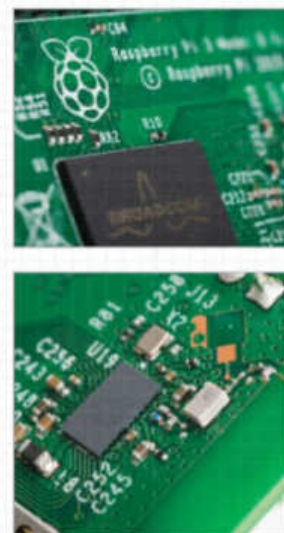
The extra performance comes at a cost, however: heat. Under heavy load, the BCM2837 runs anywhere from 83°C up to 100°C – and in testing we found our models failing to throttle when this out-of-spec high was hit, an issue that the Raspberry Pi Foundation was at the time of writing unable to replicate. With the Pi 2 topping out around the 50°C mark, that’s a major increase – and one that will only get worse if you’re housing your Pi in a case. It’s little surprise, then,

that the Pi 3 is the first model the Foundation has suggested “may need a heatsink.” Although one is not supplied, any 14-15mm² heatsink proves suitable, and can drop the temperature down to below 70°C even when used passively.

The fact that so little has changed about the design of the Raspberry Pi 3 is mostly a positive. The majority of existing accessories and add-ons are immediately compatible, though cases with poor ventilation should be used with care and there are some compatibility issues still to be ironed out with a change to the GPIO port’s UART.

One negative can be found in the USB and Ethernet ports: just like every model going back to the original Model B, these are connected to the SoC via a single shared USB channel, strangling the bandwidth available – although at least the power each port can provide has been boosted to improve peripheral compatibility.

■ **Gareth Halfacree**



Pros

Significant speed boost over earlier models; built-in wireless connectivity; near-total backwards compatibility; 64-bit potential

Cons

Significant increase in heat output; no change to the VideoCore-IV graphics processor; no software to take advantage of the 64-bit CPU

Summary

Wireless connectivity makes the Pi 3 a step up from its predecessors, even before considering the new performance highs. The new chip’s thermal performance raises concerns, though, and the new Pi has all of the same drawbacks as its predecessors, including shared bandwidth between USB and Ethernet.

8

SUBSCRIBE & SAVE UP TO **37%***



*US Subscribers save up to 40% off the single issue price.

See more at: www.greatdigitalmags.com

UX MAGAZINE FOR LESS WHEN YOU SUBSCRIBE!

Every issue packed with...

- Programming guides from Linux experts
 - Breaking news from enterprise vendors
 - Creative Raspberry Pi projects and advice
 - Best Linux distros and hardware reviewed
-

Why you should subscribe...

- Save up to 37% off the single issue price
- Immediate delivery to your device
- Never miss an issue
- Available across a wide range of digital devices



Subscribe today and take advantage of this great offer!

Download to your device now

NEXT-GEN DISTROS

Kit your desktop out with one of these radical takes on the classic distro formula

As Linux users, we're pretty fortunate when it comes to being able to switch up our desktop experience whenever we see fit. The process of switching between distros has never been easier and, thanks to the likes of VirtualBox, testing out potential distros can also be completed in a matter of minutes.

But despite the increased amount of choice in distros and the simplicity of switching between them, many Linux users still rely on the traditional powerhouses for their needs. The likes of Ubuntu, Linux Mint and Debian have long been recognised as one of the primary reasons for the year-on-year increase in Linux users around the world, offering accessibility to the OS for those making

the transition from Windows and OS X. However, the next generation of Linux distros is upon us. Promising new features and better designs, but without compromising on usability, these relatively new distros are creating quite a stir among the community – and rightly so.

Some of them are heavily forked versions of classic distros, while some bring something entirely different to the table – with varying results. We've got hands- and eyes-on with some of the most audacious and inventive distros that have landed recently, to uncover the next generation of distros all vying to take over your desktop in the near future. Perhaps they'll even change your perception of Linux as a whole.

SOLUS OS



Graphical interfaces have never looked this good

For the seasoned users among you, chances are you may have previously heard about the development of the Solus operating system – for both good and bad reasons. For those less aware of the behind-the-scenes antics, you might want to check out our full interview with its founder, Ikey Doherty, back in **LU&D 162**.

Following on from a successful launch, the project looks to offer a stable, and alternative, take on the more common GNOME-based distributions. Without doubt, Solus' most distinguishing feature is its unique graphical interface: Budgie. The Budgie desktop has been written from the ground up, enabling easy integration into other distros and is essentially an open source project in its own right.

At its core is its simplicity, which is something that Solus heavily relies on, and it pays dividends when moving around the interface. A large part of this is the Raven applet technology that provides an intuitive notification centre for desktop users, which is still fairly uncommon for most modern distros to contain. It's also highly accessible out-of-the-box, not only due to the amount of key software readily available, but also the use of 'high contrast' theming to make it the distro of choice for those with visual impairments.

Solus OS has just launched its first point release (1.1 Shannon) and with that, it has optimised, refined and tweaked Budgie to be better than ever before. For those who are after a traditional desktop experience, but don't want to compromise on how their interface looks, then you really need to download Solus as soon as possible. Official downloads can be found over at solus-project.com.



SAFETY IS KEY WITH SOLUS OS

Fewer configurations, easier installation

What will be pleasing to users testing out Solus for the first time will be the lack of multiple toolchain versions littering their site. Instead the Solus team ships only the latest build, making sure everyone can have the option to update at the same time. What this also helps with is preventing hundreds of copycat, and sometimes untested, configurations from appearing that fall outside of the

development team's control. Having a 'single download fits all' mentality can only be a good thing for both new and seasoned Linux users alike; the potential problems faced when downloading some of these untested configurations can lead to some big problems for your desktop. Thankfully, Solus gives you all the tools you need so you won't need to look elsewhere for your downloads.



Above The smart Budgie desktop has already spread to distros beyond Solus

ALPINE LINUX



The true definition of a minimal distribution

Compared to many of the distributions that we're featuring here, Alpine Linux stands out as being one of the few distros that caters for low-resource computers. Its default containers require no more than 8MB of space and its minimum installation to disk takes up a paltry 130MB of HD space. Perhaps the biggest positive from this is that you can expect lightning-fast install times. And, despite its small size, users still get a fully-fledged Linux environment, but also a set of repository packages to boot. Various binary packages have been split up and thinned out, enabling the user to have more control of what they install. There's plenty of choice here, so don't worry about Alpine being overly simplistic.

In its most recent update, it was announced that Alpine would be teaming up with Docker, again another nod to their promise of leaving a very small system footprint. Docker's containers don't require a separate, full operating system, and can instead make use of the low resource system that Alpine provides. It's a match made in heaven, if you ask us. A wide range of ISO files are readily available through the official Alpine Linux site, with a customised download available for those wanting to use it through their Raspberry Pi.



Above Alpine Linux is a minimal distro built for low resource computers

KORORA



An out-of-the-box experience that comes in different flavours



Since its inception back in 2005, Korora has undergone numerous updates, been abandoned, rescued from obscurity and has recently been revamped into something completely new. To say it has gone on a journey would be an understatement, but the finished product proves to be one of the better entry-level distros out there.

While it's essentially a remix of Fedora, Korora fills in some of the gaps that Fedora often seems to miss. It ships with a number of both open-source and proprietary applications, so that the end-user doesn't need to tweak the system or set up a third-party repository. The idea of an out-of-the-box distro has never been more apparent with the ease of use that Korora both promises and delivers.

What's even better about Korora is that it offers great customisation options to its users by supporting a number of desktop environments. Users can effortlessly switch between MATE, Cinnamon, KDE, XFCE and

Gnome environments, and then perform heavy, detailed customisations, depending on their tastes and expertise.

We should also point out the existence of the helpful Pharlap tool, used for installing third-party drivers – such as graphics cards – without the complicated process than many other distros have you follow. Korora is all about simplicity, and end users are the ones who will benefit from it.



Above Manage files, or choose to distribute them to the cloud

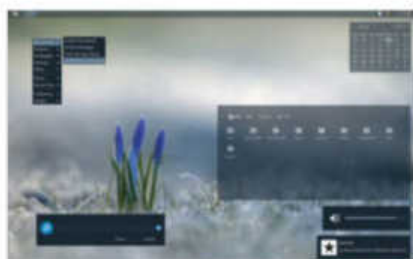
BUNSENLABS



CrunchBang lives on its community-led successor

There were a lot of users who were sad to see the demise of the CrunchBang distribution, but its successor, BunsenLabs, is really starting to take shape. It's another community-led distribution, offering both a highly customisable and lightweight Openbox desktop, with both the tint2 panel and conky system monitor readily available to use.

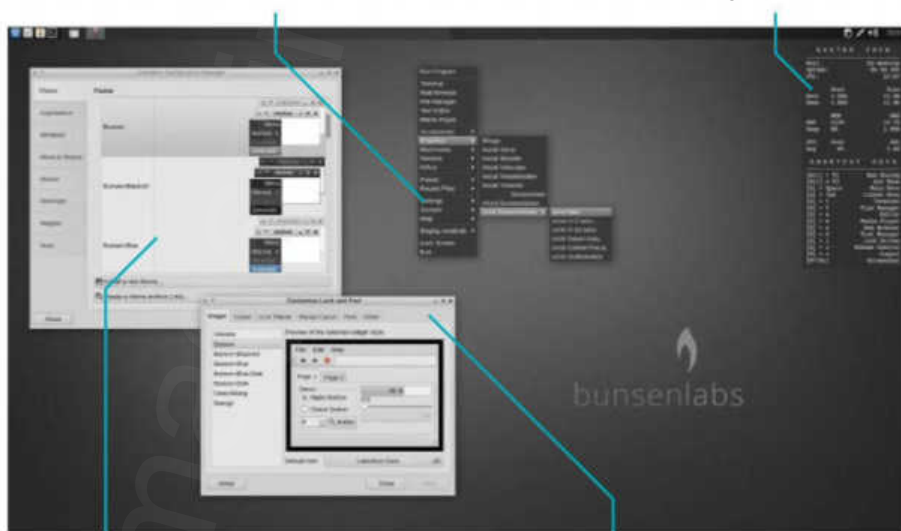
There's a range of GTK2/3 wallpapers and themes that can be used to change up the look and feel of the distribution, and several conky configurations have been implemented to cater for advanced users. Additional desktop and hardware-related packages have also been pre-installed to make it one of the few distros that offers a complete experience on the first boot.



Above BunsenLabs incorporates useful and modern-looking notifications and widgets

UNTETHERED LAUNCHER

You can open BunsenLabs' app launcher from anywhere on the desktop – just right-click to bring it up. If the feature is disabled, look for a launcher icon at the far-left of the menu bar



MONITORING WITH CONKY

Conky comes as one of the premier features of BunsenLabs. Users are able to get an overview of their system performance and the current key bindings they have in place

GTK2 THEME PACKAGE

There's a range of stylish GTK2/3 themes and wallpapers ready to be used within BunsenLabs. Similarly, users can configure various widgets, the mouse cursor and other areas of the distribution

OPENBOX WINDOW MANAGER

Openbox comes pre-configured in BunsenLabs, providing a minimal approach to window management. Openbox is one of the more customisable window managers available

NIXOS



An all-in-one solution for system configuration

System configuration is at the heart of NixOS, and for those who really want to get hands-on with their desktop, this is arguably the best distro out there. Essentially, Nix combines things like configuration rollback and package management into a single configuration file that can then be activated easily with a sole command.

The other unique feature of Nix is that each system configuration file is saved to the system and then freely available to boot until it's wiped from the system. For developers, it provides a relatively simple way of rolling back to a previous configuration when necessary.

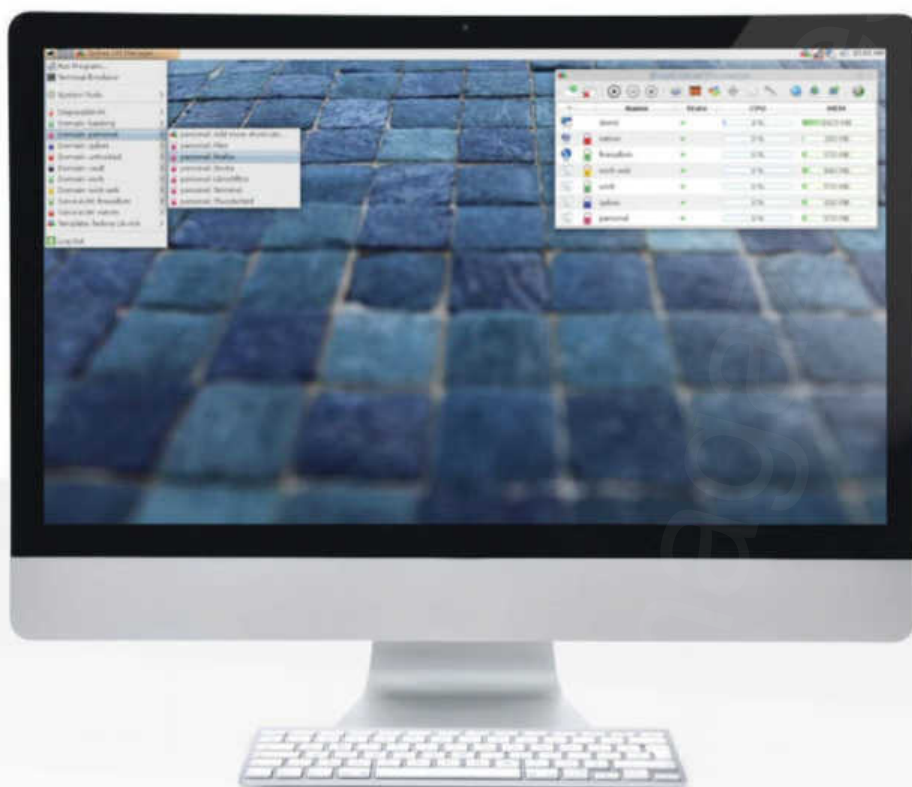
If you're able to get your head around a non-standard directory structure and the steep learning curve applied here, NixOS provides an excellent suite for DevOps and makes large-scale cloud deployments much simpler.



QUBES



Take your security game to the next level



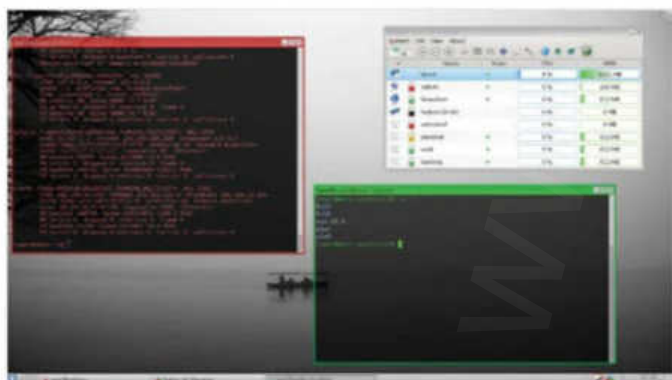
“Qubes’ main focus is centred around security by compartmentalisation”

Despite the protection we implement on our machines, there's always the concern they will be compromised. While we've encountered numerous security-orientated distros, Qubes' main focus is centred around security by compartmentalisation, which allows for different parts of your digital footprint to be isolated between different virtual machines.

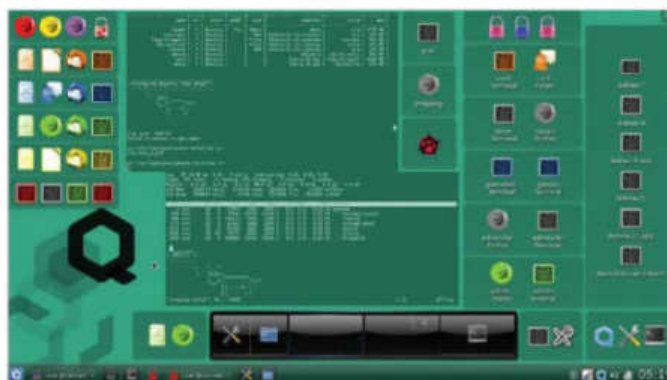
While it may seem overly complex, the premise is actually fairly simple. By splitting up your digital footprint into different VMs, essentially if one of them is compromised, there's no way for other parts of your footprint to fall victim to the same fate. These separate VMs are also implemented if you accidentally open a malicious website or download a dodgy attachment. It's all kept separate so that your digital life won't be taken down in one fell swoop.

However, virtual machines can also be compromised, right? Luckily, Qubes has taken care of that too. Where traditional VMs run under the host OS, Qubes run off the core hardware. What this means is that the attacker must be capable of breaking through the entirety of the hardware, which is vastly more difficult, rather than just the OS.

A recently implemented installer means that things like a USB VM to protect against USB-sourced malware, and network VMs to quarantine sensitive documents, will be made easily available. What Qubes is offering is a new solution to desktop security; if you care about your digital security, this is the distro you need.



Above Use the VM Manager to manage the segmentation of your system into discrete areas that cannot interact with each other



Above Here you can see some of the ways in which KDE can be customised to work with Qubes



VELTOS

Built by the people, for the people

Everything included in VeltOS has been campaigned and voted for by its community and users. That means anyone can set up a poll or discussion, and if it gets enough positive feedback, it'll be implemented as part of the next update. It's a great idea for empowering its users, and has allowed for some rather quirky features to be included. Because of the community-driven aspects of the OS, a mixture of official Arch and AUR packages fulfil most of the

default software. Most of these will instantly be recognisable as leading software in their fields, but there are a few surprise additions included for good measure.

While it's still in its early development phase, there's a lot to be excited about in VeltOS. A community-driven OS isn't an entirely new idea, but the scale at which the Velt team is putting the responsibility into the end users' hands is really quite refreshing.

PURE OS

A tweaked Trisquel-based distro for the 'free' generation

Pure OS is one of the very seldom, free software-approved Trisquel distributions, and it massively benefits from it. All of its key drivers, standard suite of software and other elements of its Gnome 3-based desktop environments have been selected based solely on free software.

Sure, it arguably lacks the finesse of some of the more fleshed-out distributions out there, but there's definitely an audience out there for

Pure OS, especially those who want each and every part of their desktop experience to be completely and utterly free – and sourced from the right places.

We also tip our hats to the included PureBrowser, which is a heavily modified version of Firefox, with some security plug-ins; namely HTTPS-Anywhere and Privacy Badger pre-installed. It's easy to use, and we'd love to see it added to other distributions.

NEXT-GEN FOSS

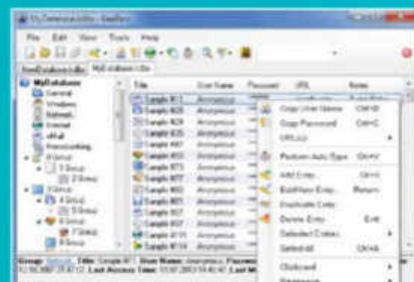
Pack your desktop with some of the best FOSS out there

PIDGIN IM

As far as IM clients go, Pidgin IM has long been one of the better ones around. As well as one-on-one chats, it also provides support for multi-chat, enabling users to sign into multiple accounts simultaneously.

KEEPASS

Keeping tabs on your multitude of accounts isn't easy, so having a piece of software that can do the work for you is invaluable. KeePass uses AES encryption to keep your log of passwords and accounts completely safe.



Above KeePass can safely store sensitive documents and SSH keys as well as passwords

DARIK'S BOOT AND NUKE

If you're ever in the position of needing to wipe hard disks in a matter of minutes, DBAN should be your FOSS of choice. It provides all the necessary tools for bulk and emergency data destruction.

ENDIAN FIREWALL COMMUNITY

Endian gives users the necessary tools to turn their system into a security appliance. It contains a packet inspection firewall, spam filtering and a hassle-free VPN that's based closely on OpenVPN.

SYNFIG

Synfig is a powerful, vector-based 2D animation package that aims to help digital artists produce high-quality 2D animations from scratch. The learning curve is steep, but it's managed to get quite a following already.

COREOS

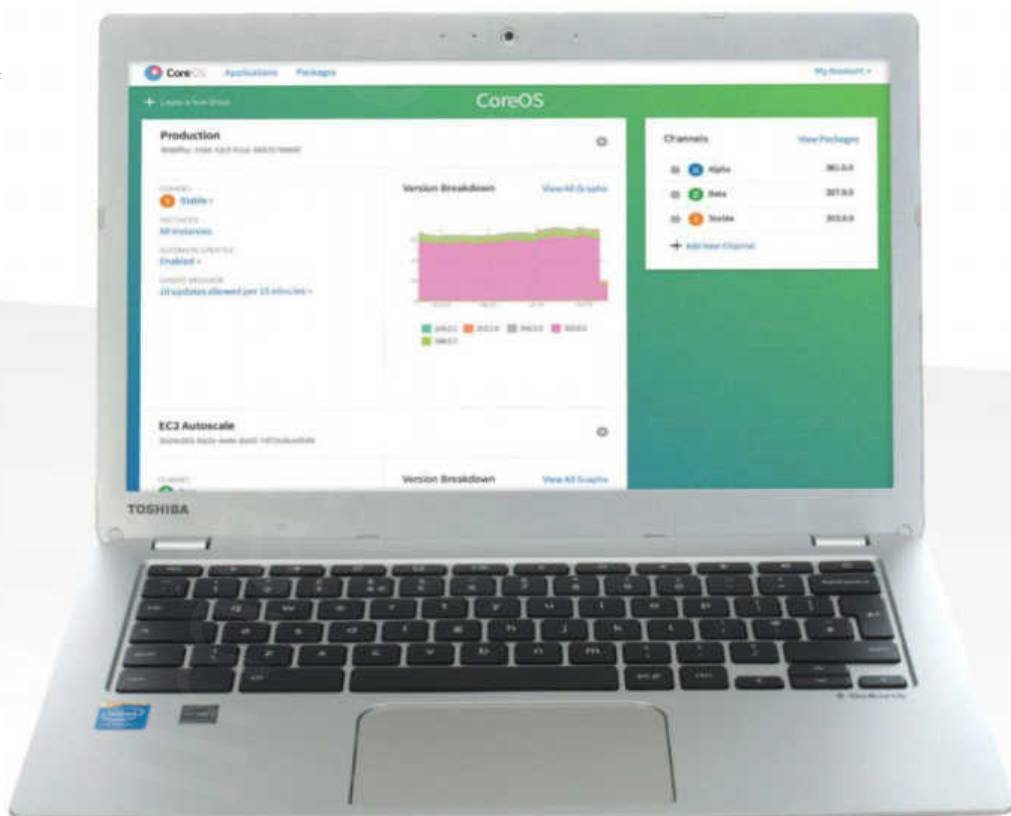


Make your desktop work for you with the help of CoreOS

When you first get hands-on with CoreOS, you can be forgiven for thinking it's a bare bones distribution, but that turns out to be one of its greatest strengths. The positive aspect of this minimal approach is that CoreOS is capable of trying out different deployments and dependencies, with the end user having a distro that's easier to scale and manage.

Perhaps CoreOS's core selling point is that it doesn't ship with a package manager, and instead relies on containers to help package applications and dependencies, the most prominent being Docker. Users can construct these containers for each service you want on your machine (caching, database), use the fleet tool to start them and then connect them together with etcd.

Its other big selling point is through its update system with its root partition scheme, FastPatch. Instead of updating through packages, CoreOS is updated by a single unit, allowing for exceptionally quick and timely downloads. Essentially it minimises the overall complexity of the update process by compartmentalising each entity. While it probably isn't suited for those who are new to Linux, it's one of the few lightweight distributions out there where the control is fully in the end user's hands.



CUB LINUX



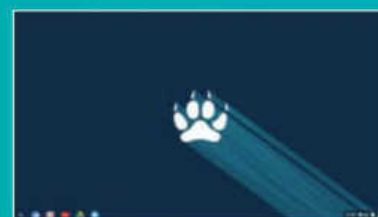
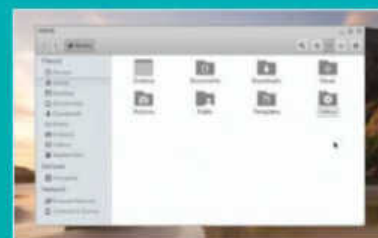
Chromium and Ubuntu in one helpful package

We've already raved about how great Chromium is, but when mixed with Ubuntu, it gets even better. Cub Linux is a unique mix of the best aspects of the Chromium browser, namely its speed and web apps, with the best of what Ubuntu can offer. The end result is a combination of both speed and power.

While Cub Linux doesn't use any components of actual Chromium OS, it does

use a variety of open source products that look and act the same as the browser, but without the over-reliance on solely using web apps. With the power of Ubuntu behind it, users can now customise their Chrome OS experience with a myriad of desktop applications, something that wasn't possible before.

If you want the power of the Google ecosystem, but without being directly tied into the cloud with the help of desktop applications, Cub Linux is a worthy alternative to your standard Ubuntu experience.



REMIX OS



The full Android experience on any desktop

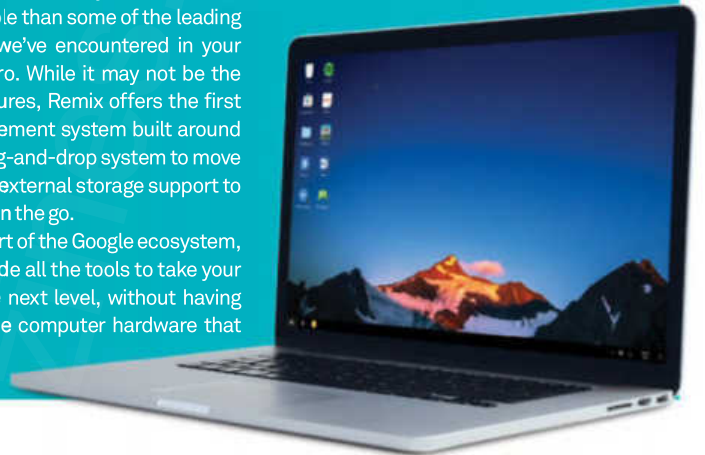
Based on Android-x86, Remix OS is one of the first mainstream attempts at getting Android on to either a PC or Linux desktop. It adds pretty much everything you'd have previously encountered with Android in general, but with the added support of a window manager, taskbar and start menu.

Android has always been about notifications, and the way Remix has added into its OS not only looks great, but is incredibly useful. Notifications are split between apps, core settings and

messages, allowing for easy control. It's noticeably more usable than some of the leading notification centres we've encountered in your traditional Linux distro. While it may not be the most exciting of features, Remix offers the first advanced file management system built around Android. It uses a drag-and-drop system to move files around, and has external storage support to hook up a hard drive on the go.

If you're already part of the Google ecosystem, then Remix can provide all the tools to take your Android usage to the next level, without having to compromise on the computer hardware that you're using it on.

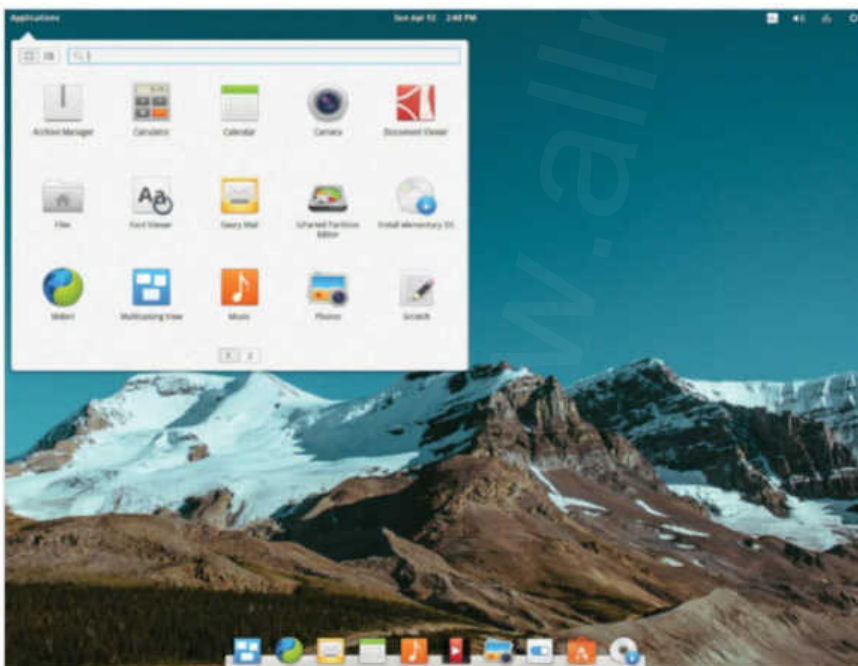
“It uses a drag-and-drop system to move files around”



ELEMENTARY OS



Wait, this isn't OS X?



Above An OS X-style dock sits at the bottom of the distribution

A large part of why Elementary has gained a slight cult following since its inception a couple of years ago is due not just to its ease of use, but also its design ethic. For those moving from Windows, or even OS X, Elementary offers a recognisable and functional alternative to the standard proprietary operating systems. Think of it as Linux, but a very pared-down version of it with a focus on usability.

Simplicity is key here and a large portion of it centres around the multitasking aspects of the desktop environment. We've seen other distributions use the multi-window feature previously, but Elementary's implementation is slightly different. It uses hot spots on your desktop for effortless switching between windows, most of which can be activated with either a click of the mouse, or a customisable keyboard shortcut.

And if that wasn't enough for you, then the team behind the OS has also created a suite of pre-installed software, custom-made for their desktop environment. Again, it's all centred around simplicity and ease-of-use. This perhaps won't tick the right boxes for advanced users, but for those making the transition across to Linux for the first time, it's a great place to start.

PLASMA OS

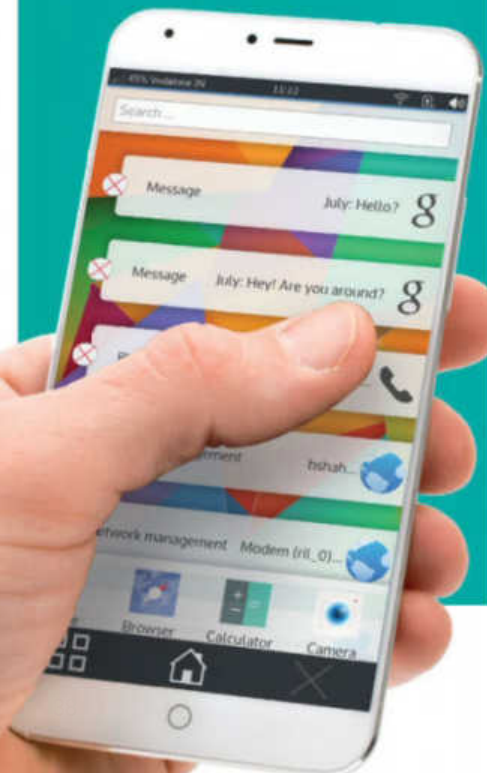


A radical alternative to Ubuntu Touch

Plasma is open source in every sense of the word. It can be modified, distributed and re-used without *any* restrictions in place. There's a buzzing community already behind it that's already helped spur on a number of key developments in the early stages.

Unlike many of its competitors, Plasma Mobile can support different types of apps. So as well as the staple Android apps, there's also support for apps developed by GTK, Ubuntu Touch and Sailfish OS. An all-encompassing ecosystem is a great way of getting more people interested in a fledgling OS such as this.

Personalisation plays a big role in the mobile OS, enabling users to tinker with every aspect of their smartphone experience, as they would with a desktop. With a frenzy of developers working in the background, and some great incentives out there for others to get involved, Plasma Mobile could play a major part in your next smartphone experience.



SAILFISH OS



The truly independent mobile operating system

The core of Sailfish OS is built in the same vein as a classic Linux distribution, based on the Mer Project – a mobile-optimised distro. While there are some noticeable similarities between the look and feel of Sailfish and Android, it actually relies more heavily on user participation for end results. Development kits for OEMs, chipset providers and operators are all readily available for download, with the ultimate aim to make Sailfish as cohesive as possible.

A large portion of Sailfish is designed to be modular and easily supportive of multiple hardware targets. Any third-party elements are used as external features for the core OS, enabling experimentation with different areas of the device to figure out what works best together. While the distro is still in the early stages of development, there are already offshoots of the core OS available for a myriad of devices, so it's worth experimenting with different versions.



TIZEN



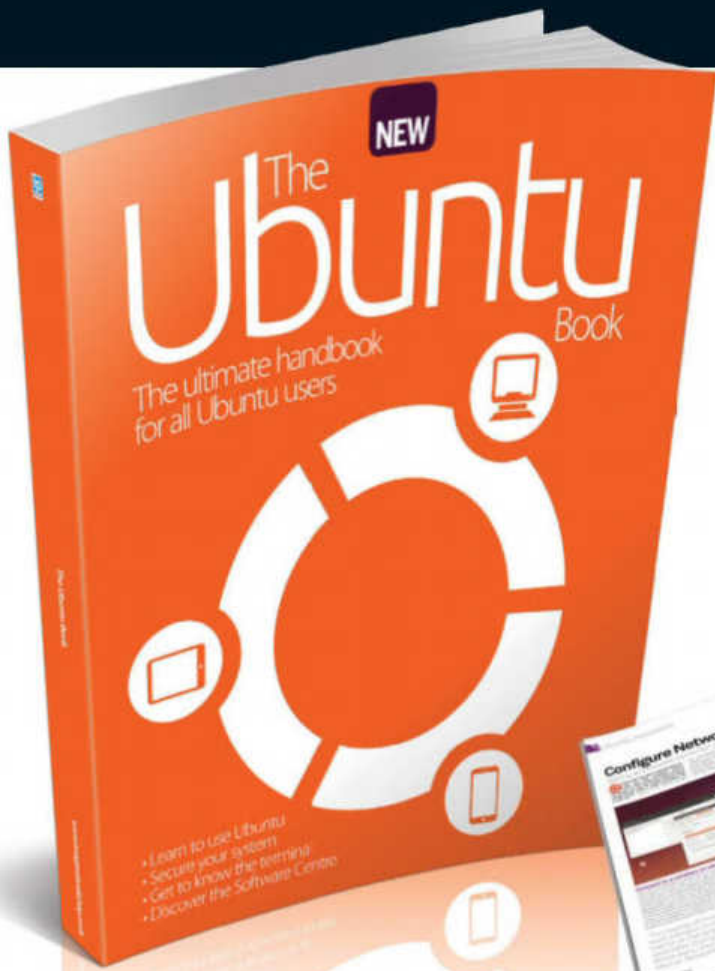
The Linux Foundation's pet mobile project has come of age

You know when The Linux Foundation partners with the likes of Samsung and Intel, you're in for something exciting. Tizen is their joint vision of what the future of mobile operating systems should look like. Being built completely from the ground up has enabled the OS to be ported on to numerous devices. As well as smartphones, there's current support for Tizen on a number of smartwatches, tablets and even house appliances. Due to the high flexibility of the OS, a device manufacturer, or even network,

can modify and create their own fork of Tizen, depending on what they need from it. In essence, this enables manufacturers to mould Tizen around a specific hardware feature.

For app developers and software vendors, the OS offers the winning combination of both native app development and high levels of HTML5 support. It's also been targeted as one of the leading mobile platforms for the next gen of smart devices. Expect to see Tizen go far beyond your smartphone in the years to come.

From the makers of **LinuxUser** & Developer

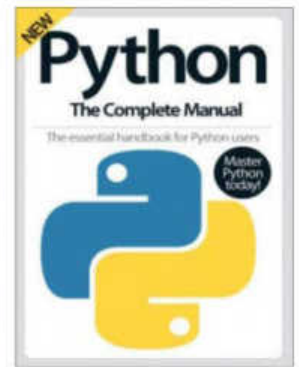
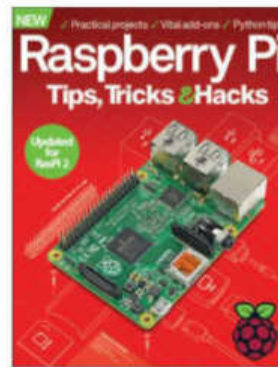
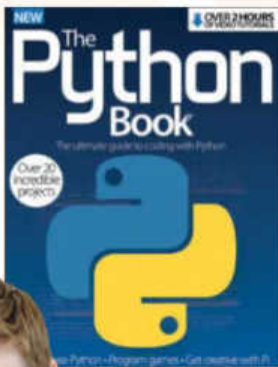


The Ubuntu Book

Whether you're new to Ubuntu or looking to get even more from the operating system, this is the book for you. Discover the best features of Ubuntu's default software, and the essential FOSS apps for you to download from the Software Centre. This is the only book you need to master Ubuntu.



Also available...



A world of content at your fingertips

Whether you love gaming, history, animals, photography, Photoshop, sci-fi or anything in between, every magazine and bookazine from Imagine Publishing is packed with expert advice and fascinating facts.



BUY YOUR COPY TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com





Mihalis Tsoukalos

is a Unix administrator, a programmer (for Unix and iOS), a DBA and also a mathematician. He has been using Linux since 1993

Resources

Text editor
GCC compiler
Go compiler
Rust compiler
Perl interpreter

Systems programming

Advanced file input/output

We return to file I/O this month with an in-depth look at the more advanced techniques you can employ

Advanced file I/O means that you can do all usual file I/O operations with extra powers such as record locking, nonblocking I/O, I/O multiplexing, asynchronous I/O, memory mapped I/O, and scatter read and gather write. Don't worry; we will take you through everything you need to get started.

This tutorial will present the system calls and explain the techniques that will allow you to perform advanced file I/O; it will be the most challenging subject you have seen in the Systems Programming series. Most of the code in this article will be written in C but you will also see examples in Go, Perl and Rust.

Please bear in mind that the concepts and principles of advanced file I/O are more complex than standard file I/O. When in doubt, look at the source code or try to write your own programs – this will definitely solve any ambiguities and make things clearer for you.

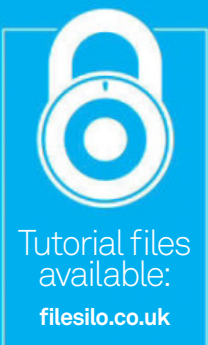
Blocking and nonblocking I/O

There exist two categories of system calls: the usual ones like `read()` and `write()`, which can block forever,

and the nonblocking ones, which return with an error message indicating that the operation was not completed and could have been blocked.

As you might have guessed, advanced I/O uses the nonblocking kind of system calls. The key feature of nonblocking system calls is that they return immediately, which means that a process does not have to wait for something to finish. Blocking file I/O is not the desired kind of file operation because it can make a program wait forever and never terminate – the only way to stop a blocked process is with the help of a signal; you can either press `Ctrl+C` or use the `kill` utility to send a proper signal. Both scenarios should be avoided in applications used in production!

Another scenario where a program can wait forever is in a deadlock situation; yet another area is network programming, which is related to file I/O, because it is most likely to wait “forever” due to a network-related problem than a problem on your local filesystem – those of you who remember having NFS-related problems will understand!



```

1  #include <unistd.h>
2  #include <fcntl.h>
3  #include <stdio.h>
4
5  int main(int argc, char const** argv)
6  {
7      char buf[20];
8      int n = read(0, buf, 4);
9      if(n > 0)
10         printf("You typed: %s", buf);
11         return 0;
12 }

```

Left This is an example of user blocking, where the code will endlessly wait for user input

A simple example

Let's start off with a simple example. The image above shows the source code of `simple.c`, which presents a file I/O operation that can be blocked because of the user. The C code that allows the program to block is the following:

```
int n = read(0, buf, 4);
```

The program waits for input from the user – if the user does not press the Enter key, the program will wait forever, which is not considered a good practice. Although the blocking happens because of the user, it is still considered blocking.

If you change the `simple.c` code as the output of the `diff(1)` utility shows, then it cannot be blocked and will always terminate even without completely finishing its job:

```

$ diff simple.c simpleNB.c
7a8,9
> fcntl(0, F_SETFL, fcntl(0, F_GETFL) | O_NONBLOCK);
> sleep(4);

```

As you can see, the overhead for making `simple.c` to behave in a nonblocking way is small – if you want to truly improve the program you might need to write even more code, but at least `simpleNB.c` will not block your system.

Some useful system calls

Almost all techniques presented here need some special system calls to be implemented. This part will present some of the most important system calls you need to be aware of. You already know about the `fcntl(2)` system call (`simpleNB.c`), which can do file control in general, including non-blocking file I/O. It is now time to fully explain the following call to `fcntl(2)`:

```
fcntl(0, F_SETFL, fcntl(0, F_GETFL) | O_NONBLOCK);
```

First, the meaning of `F_SETFL`, `F_GETFL` and `O_NONBLOCK` should be clarified. The `F_GETFL` command tells `fcntl(2)` to return the current value of file status flags. The way the external `fcntl(2)` call is defined makes the internal `fcntl(2)` call return the current status flags and add the `O_NONBLOCK` flag to the existing ones. It then executes the `F_SETFL` command that tells `fcntl(2)` to set the file status flags to the new value that is specified by the third argument of the external `fcntl(2)` call.

The `O_NONBLOCK` flag dictates that a file descriptor is to be set in non-blocking mode, which is specified as the first parameter of the `fcntl(2)` function.

As you can understand, the `F_SETFL` command makes `fcntl(2)` require three arguments, whereas the `fcntl(2)` call with

Balancing pros and cons

The main disadvantage of all advanced file I/O techniques is that they need additional code, which means that you might create more bugs; other than this there are only advantages. What is really important is deciding whether the extra effort can be justified by the benefits they might bring. The key here is to compare the speed of normal file I/O with the speed of an application that uses an advanced technique before deciding.

It is expected that you might never use some of them; memory mapped I/O is a specialised technique that can be used for dealing with the shared memory of related processes, but might not be appropriate for reliably and quickly copying a binary file. Another thing you must consider is that most of the time you will need one or at most two of these techniques at the same time. Please note that Scatter-Gather I/O can be done with multiple calls to read and write but this is inefficient because of the overhead associated with each kernel call.

■ Advanced I/O in Go

You can do many things with Go, including the implementation of the techniques discussed in this tutorial. The `advIO.go` code shows how to perform memory mapped File I/O in Go. The program requires the "unsafe" Go package because it recasts pointers – this means that if you do not handle pointers correctly, it will crash. On the other hand, if you do handle pointers properly, your programs will run fast.

The most interesting part of `advIO.go` is the following because it is where the mapping takes place:

```
mmap, err := syscall.Mmap(int(map_file.Fd()), 0, int(t),
syscall.PROT_READ | syscall.PROT_WRITE, syscall.MAP_SHARED)
if err != nil {
    fmt.Println(err)
    os.Exit(1)
}
map_array := (*[n]int)(unsafe.Pointer(&mmap[0]))
```

The full Go code of `advIO.go` can be found in your resources pack. If you run it, you are going to get the following output, which is what is going to be written to the binary file:

```
$ go run advIO.go
[0 1 4 9 16 25 36 49 64 81]
```

the `F_GETFL` command needs only two arguments. This is because `fcntl(2)` accepts a variable number of arguments just like `printf()` and `scanf()` do.

The following output from the `fcntlExample.c` program illustrates the value that the `F_GETFL` command returns on a Debian Linux system:

```
$ ./fcntlExample
Before: fd 0 flags: 8002
After: fd 0 flags: 8802
```

So, the flags have no recognisable meaning unless you know what you are looking for. A big advantage of `fcntl(2)` is that it is largely standardised and is available across many different platforms.

A small part of the `select(2)` man page can be seen below. The `select(2)` system call allows a program to supervise multiple file descriptors. The timeout in the `select(2)` call is defined with a struct `timeval`.

The `pselect(2)` system call is almost identical to `select(2)` apart from three differences: first, the timeout is defined with a `timespec` structure; second, `pselect(2)` cannot update the timeout value, whereas `select(2)` can; and third, `pselect(2)` has a `sigmask` argument whereas `select(2)` does not.

Right We use the `select(2)` system call to monitor file descriptors to see when they are ready for an I/O operation

```
2, mtsouk@mail: ~ (ssh)
SELECT(2) Linux Programmer's Manual SELECT(2)

NAME
    select, pselect, FD_CLR, FD_ISSET, FD_SET, FD_ZERO - synchronous I/O multiplexing

SYNOPSIS
    /* According to POSIX.1-2001 */
    #include <sys/select.h>

    /* According to earlier standards */
    #include <sys/time.h>
    #include <sys/types.h>
    #include <unistd.h>

    int select(int nfds, fd_set *readfds, fd_set *writefds,
               fd_set *exceptfds, struct timeval *timeout);

    void FD_CLR(int fd, fd_set *set);
    int  FD_ISSET(int fd, fd_set *set);
    void FD_SET(int fd, fd_set *set);
    void FD_ZERO(fd_set *set);

    #include <sys/select.h>

    int pselect(int nfds, fd_set *readfds, fd_set *writefds,
               fd_set *exceptfds, const struct timespec *timeout,
               const sigset_t *sigmask);

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

pselect(): _POSIX_C_SOURCE >= 200112L || _XOPEN_SOURCE >= 600
```

The difference between `timeval` and `timespec` structures is that the former uses seconds and microseconds, whereas the latter uses seconds and nanoseconds. As you can see, a `timespec` structure can therefore be more accurate than a `timeval` structure.

Put simply, both system calls allow a program to watch over multiple file descriptors and wait until one or some of them become available for some kind of I/O operation. As a result, the programmer does not have to write any code to individually check the availability of each open file descriptor. In case of error, both functions return -1.

The `poll(2)` system call is similar to `select(2)`; it waits for some event on a file descriptor. Similarly, the `ppoll(2)` system call is analogous to `pselect(2)`. Generally speaking, whenever you feel the need to clarify or remember the purpose of a parameter or an entire system call, its man page should be the first place you visit for information.

Two simple examples

It is now time to present some simple C code. The following code, which is included in the `select(2)` man page, is a small part of `selectEx.c` and shows how `select(2)` is used:

```
tv.tv_sec = 5;
tv.tv_usec = 0;
retval = select(1, &rfd, NULL, NULL, &tv);
```

Compiling `selectEx.c` and executing it without giving any input, produces the following output:

```
$ ./selectEx
No data within five seconds.
```

The following C code, part of `pollEx.c`, illustrates the use of the `poll(2)` function:

```
retval = poll(ufds, 1, 2500)
```

The `pollfd` structure is used by `poll(2)` to keep the list of file descriptors. You can see that `poll(2)` takes three parameters, and returns an integer value which is either -1 in case of error or the number of descriptors that are ready for I/O. The second parameter is the number of `pollfd` structures, whereas the third parameter is the timeout. Executing `pollEx.c` generates the following kind of output:

```
$ ./pollEx
23
0: Input given
43
1: Input given
2: No input given!
```

Both `select(2)` and `poll(2)` functions implement a synchronous mode of communication, because the application itself has to ask in order to communicate and deal with data.

Deadlock detection

You already know that deadlocks should be avoided. However, there are times where something might go wrong and a surprise deadlock happens. The solution to this problem is deadlock detection. The code of `simple.c` also shows a case where a deadlock might happen, but this is due to a user not pressing the Enter key. The nastiest deadlocks usually happen when process A waits for a resource that is held by process B,

```
1  #include <fcntl.h>
2  #include <errno.h>
3  #include <stdio.h>
4  #include <unistd.h>
5
6  int lockFile(int fd, int limit)
7  {
8      int try = 0;
9      struct flock lck;
10     lck.l_type = F_WRLCK;
11     lck.l_whence = 0;
12     lck.l_start = 0L;
13     lck.l_len = 0L;
14
15     for (try = 0; try < limit; ++try)
16     {
17         if ( 0 == fcntl(fd, F_SETLK, &lck) )
18             break;
19         if ((errno != EAGAIN) && (errno != EACCES))
20             break;
21     }
```

Left This enables us to set a lock for a particular file

Right It's worth reading the descriptions of the various asynchronous I/O system functions

```
2. mtsouk@mail: ~/docs/article/working/SPAdvIO.LUD164/code (ssh)
AIO_READ(3) Linux Programmer's Manual AIO_READ(3)

NAME
    aio_read - asynchronous read

SYNOPSIS
    #include <aio.h>

    int aio_read(struct aiocb *aiocbp);

    Link with -lrt.

DESCRIPTION
    The aio_read() function queues the I/O request described by the buffer pointed to by aiocbp. This function is the asynchronous analog of read(2). The arguments of the call

        read(fd, buf, count)

    correspond (in order) to the fields aio_fildes, aio_buf, and aio_nbytes of the structure pointed to by aiocbp. (See aio(7) for a description of the aiocb structure.)

    The data is read starting at the absolute file offset aiocbp->aio_offset, regard-
```

while process B waits for the resource that is held by process A. The `deadlock.c` file showcases this effective technique where file locking (`fd`) takes place using the following:

```
fcntl(fd, F_SETLK, &lk)
```

When the lock is successfully placed, the `lockFile()` function returns 0. Executing `deadlock.c` will produce the following output:

```
$ ./deadlock deadlock.c
File deadlock.c is successfully locked!
```

Please note that you will need to make a copy of the `deadlock.c` executable and execute it. This is needed because only the first instance you run can actually lock the file!

```
$ ./anotherDeadlock deadlock.c
File deadlock.c cannot be locked!
```

Be sure to download and read the source code of `deadlock.c`.

Asynchronous I/O

In the asynchronous form of communication, the read and write operations take place in the background while the program continues its execution.

The POSIX standard defines an asynchronous I/O interface (AIO) control block to perform I/O operations using the `aiocb` structure that is defined as follows:

```
struct aiocb {
    int      aio_fildes;    /* File
                           descriptor */
    off_t    aio_offset;    /* File offset */
    ...
};
```

```
volatile void *aio_buf;    /* Location of
                           buffer */
size_t      aio_nbytes;    /* Length of
                           transfer */
int         aio_reqprio;    /* Request
                           priority */
struct sigevent aio_sigevent; /* Notification
                           method */
int         aio_lio_opcode; /* Operation to
                           be performed; lio_listio() only */
};
```

The functions used for programming the POSIX asynchronous I/O interface include: `aio_read(3)`, `aio_write(3)`, `aio_error(3)`, `aio_return(3)`, `aio_suspend(3)`, `aio_cancel(3)` and `lio_listio(3)`. Please visit the man page of `aio(7)` for more information. In the image above, you can see a small part of the man page of the `aio_read(3)` function.

Other ways to implement asynchronous file I/O is with the help of signals or by using threads. Talking more about asynchronous I/O is beyond the scope of this tutorial.

Memory mapped file I/O

Memory mapped I/O is a specialised technique that connects a file on disk with a buffer in memory in such a way that when you get bytes from the buffer, the matching bytes of the file are actually being read and used. Analogously, when you write in the buffer, the data is automatically written to the physical file. Please note that the buffer need not have any memory space allocated to it because the memory space comes with the real file that is associated with the buffer!

The main system call for memory mapped file I/O is `mmap(2)`, which maps pages of memory and takes six

```

8  int main(int argc, char const *argv[])
9  {
10     char *buffer;
11     int size;
12     struct stat s;
13
14     if ( argc != 2)
15     {
16         printf("You need to have exactly one argument\n");
17         return -1;
18     }
19
20     const char * filename = argv[1];
21     int fd = open(filename, O_RDONLY);
22
23     // Get the size of the text file
24     int status = fstat(fd, &s);
25     size = s.st_size;
26
27     buffer = (char *) mmap (0, size, PROT_READ, MAP_PRIVATE, fd, 0);
28     for (int i = 0; i < size; i++)

```

Left Using memory-mapped file IO can be much more efficient when reading and writing

parameters. You will have to open the file in advance because one of the parameters in `mmap(2)` is the file descriptor of the true file. Another argument specifies what kind of access is allowed to the memory-mapped region, and has four possible values which you can combine: `PROT_READ`, `PROT_WRITE`, `PROT_EXEC` and `PROT_NONE`.

The image above shows some of the C code of the `memory.c` file. As you can see, the code is relatively easy to read, but this is not usually the case with memory mapped I/O because of the various offset variables that are needed in order to access big files fast.

Executing `memory.c` prints the following output:

```

$ cat testFile.txt
1
Line 2
Line 3
$ ./memory testFile.txt
1
Line 2
Line 3

```

The biggest benefit from Memory Mapped file I/O is increased I/O performance, especially when you are dealing with large files.

The `readv` and `writv` system calls

To wrap things up, let's take a look at the `readv(2)` and `writv(2)` functions, which implement scatter read and gather write file I/O, where the system reads into multiple buffers and writes from multi buffers. Both operations are atomic – this means that if you call `writv(2)` then all data in the vector will be written in one and only one step; therefore, no other writes by other

processes can happen in between. This is very important because many critical applications, including database servers, require this kind of atomicity.

There is a small example included in your resources pack (`rwV.c`) where you can see both `readv(2)` and `writv(2)` in action. As you can see in the C code, an `iovec` structure is used in both functions. The `iovec` structure is defined as follows and keeps track of the various buffers used:

```

struct iovec {
    char    *iov_base; /* Base address. */
    size_t  iov_len;   /* Length. */
};

```

Compiling `rwV.c` and executing it produces the following output:

```

$ ./rwV
Wrote 37 bytes
Read 37 bytes
0: Writing from
1: Line 1 using
2: writv.
$ cat testFile
Writing from Line 1 using writv.

```

It is really important that the three buffers used in the `readvEx()` function have the correct size in order to read the right parts of the file. As you can see, both writing and reading from `testFile` happened using a single function call!

Advanced file I/O is a difficult subject that you need to study carefully. So, before you use any of the presented techniques, make sure that you have written lots of examples and small projects to familiarise yourself with them. ■

Use Hadoop data via HBase



Ashish Sinha

is an author, developer, teacher, blogger and true open source enthusiast at heart. With five years of experience as a developer in a MNC, Ashish enjoys exploring new open source technology and tools

Resources

Hadoop
bit.ly/20m1Z3h

HBase
hbase.apache.org

Eclipse
eclipse.org

Hadoop is for batch processing, but for real-time access of data, you should use HBase

Big data – measured in terabytes/petabytes – cannot be stored and processed with traditional databases, so it was only a matter of time before something like Hadoop came into existence. Hadoop uses HDFS (Hadoop-distributed file system) and MapReduce to store and manipulate this data. But the problem with Hadoop was that it is only applicable for batch processing. For real-time read/write access to very large data sets, HBase emerged.

HBase is an open source, column-oriented database built on top of HDFS. The HBase project was started by Chad Wand and Jim Kellerman at Powerset and was modelled after Google's Big Table. HBase is well suited for sparse data sets, which are very common in big data. Moreover, it provides APIs that enable development in different programming languages, such as Java. The key factors that make HBase useful are that it is horizontally scalable, it can work on single/multiple nodes, and that it can be easily integrated with the MapReduce framework. In this article we will learn how to install HBase and access the data stored in HBase using a Java program.

HBase data model

Applications that use HBase store data in labelled tables containing rows and columns. The intersections of rows and columns are versioned, and by default their version is timestamp

auto-assigned by HBase at the time of cell insertion. Row columns are grouped together into column families; so every column following the first one (the row key). The table's column family must be specified as a part of the table schema definition, but new column family members can be added on demand.

The table is lexicographically sorted on row keys; row keys being a byte array, so anything ranging from string to byte to longer – even serialised data structures can serve as row keys. The coordinates used to identify data in a HBase table are a) rowkey, b) column family, c) column qualifier, and d) version. HBase doesn't support the indexing of other columns in the table, however there are several strategies for supporting the types of query that secondary indexes provide.

Region and locking

In HBase, when the table grows in size it is automatically segmented into regions – regions being the units that get distributed across the HBase cluster. A region is denoted by the table it belongs to, with its first row inclusive and the last row exclusive. Initially the table comprises a single region, but as that table grows and crosses the configurable size threshold, it splits into two regions of approximately equal size. A table that is too big for a single server is stored in a cluster of servers, with each node hosting a subset of the table's total region.

Right HBase entries are timestamped, so you can get versioning with your stored databases

```

training@localhost:~$ hbase shell
14/02/21 10:56:11 WARN conf.Configuration: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.92.1-cdh4.1.1, rUnknown, Tue Oct 16 11:44:19 PDT 2012

hbase(main):001:0> scan 'table'
ROW                                COLUMN=CELL
row1                                column=colf:a, timestamp=1456069958286, value=value1
row1                                column=colf:b, timestamp=1456070009500, value=value2
row2                                column=colf:c, timestamp=1456070032101, value=value3
2 row(s) in 0.4930 seconds

hbase(main):002:0> get 'table','row1'
COLUMN                                CELL
colf:a                                timestamp=1456069958286, value=value1
colf:b                                timestamp=1456070009500, value=value2
2 row(s) in 0.0260 seconds

hbase(main):003:0>
  
```

HBase uses ACID (atomicity, consistency, isolation, and durability) properties. When a transaction like **put** or **delete** starts, HBase retrieves the next highest transaction number, called WriteNumber. The flow of this transaction in HBase is: 1) lock the row(s) to guard against concurrent writes to the same row(s); 2) retrieve the current WriteNumber; 3) apply changes to the WAL (Write Ahead Log); 4) apply the changes to the Memstore (using the acquired WriteNumber to tag the KeyValues); 5) commit the transaction; 6) unlock the row(s).

In the case of reading data, when a transaction like **scan** or **get** starts then HBase retrieves the transaction number of the last committed transaction, called ReadPoint. The flow here is: 1) open the scanner; 2) get the current ReadPoint; 3) filter all scanned KeyValues with a memstore timestamp greater than the ReadPoint; 4) close the scanner.

Implementation

HBase is comprised of a HBase master node orchestrating a cluster of one or more region server workers. The responsibility of HBase master is bootstrapping the install, assigning regions to registered region servers, and recovering region server failures, whereas the region servers hold zero or more regions and are there to receive read/write requests. Region servers also manage the splitting and inform the HBase master node about the new regions created.

The HBase cluster is managed by ZooKeeper, which hosts vitals such as the location of HBase metacatlog table and the address of the current cluster. ZooKeeper provides a set of tools to build a distributed application that can safely handle partial failure; plus, it can be used as a rendezvous mechanism so that the different processes can interact with each other. All the partitions of regions are mediated through ZooKeeper, which keeps the assignment transactions so that if the server crashes mid-assignment, the recovery can pick up on the assignment where the crashed server left off. Region server worker nodes are listed in the HBase conf/regionserver file – to run the remote commands, an SSH-based mechanism is used.

Installation

The prerequisite for HBase is that Java and Hadoop must be installed across the nodes of the cluster and that the nodes must have a passwordless SSH connection. HBase can be installed in three modes: standalone, pseudo-distributed and fully distributed. It can be downloaded from <http://www.interior-dsgn.com/apache/hbase/stable>.

The standalone mode uses a temporary directory on the local filesystem. To set it up, first go to the conf folder, open the file hbase-env.sh and set the JAVA_HOME path here by adding the line below:

```
export JAVA_HOME=/usr/lib/jvm/java-1.7.0
```

Now open the hbase-site.xml file present in the conf folder, and under the configuration tab specify the folder containing HBase and the ZooKeeper directory, as shown:

```
<configuration>
<property>
```



Above Here is the HBase console in action –pretty similar to Hadoop on the command line

```
<name>hbase.rootdir</name>
<value>/home/hadoop/HBase/HFiles</value>
</property>
</property>
<name>hbase.zookeeper.property.Zookeeper</name>
<value>/home/hadoop/zookeeper</value>
</property>
</configuration>
```

This completes the installation. Now run the **start-hbase.sh** script present in the bin folder to start the hbase shell.

For pseudo-distributed mode, there are a few differences. In hbase-site.xml, add the below lines in the property tag to specify the mode in which HBase should be run and the URL of the HDFS instance.

```
<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:8001/hbase</value>
</property>
```

To start HBase, you must first start the Hadoop file system using the commands:

```
cd $HADOOP_HOME/sbin
start-all.sh
```

Browse through to the HBase root directory's bin folder and then start HBase:

```
cd /usr/local/HBase
./bin/start-hbase.sh
```

Now start the master and region, which are present in the same directory:

```
./bin/local-master-backup.sh start 2 (number
signifies specific server.)
./bin/local-regionserver.sh start 5
```

HBase vs Map Reduce

HBase uses HDFS in a very different form from what MapReduce does. In MapReduce, HDFS files are opened with their content streamed through a map task and then closed, whereas in HBase the data files are opened on cluster startup and are left open so that we avoid the cost of opening the files on every read/write.

Finally, start HBase:

```
cd bin
./hbase shell
```

Cloudera provides a virtual machine that comes with pre-installed software like Hadoop, HBase, Oozie, Zookeeper, etc. It can be downloaded from http://www.cloudera.com/downloads/quickstart_vms/5-5.html.

HBase read

A read in HBase is reconciled between the Hfiles, MemStore and BlockCache, where the BlockCache is used to keep frequently accessed data from the HFiles in memory so as to avoid disk reads. It keeps data in the form of blocks, as units of data that HBase reads from the disk in a single pass, and each column family has its own BlockCache.

The files in HBase are stored in HFile format – HFiles belong to a column family (one MemStore per column family) and a column family can have multiple HFiles, but the reverse isn't true. In the case of HFiles, they are laid out as a sequence of blocks plus an index over those blocks. This means that reading a block from HBase requires only looking up that block's location in the index and retrieving it from disk. Reading a row from HBase requires first checking the MemStore, then the BlockCache, and finally HFiles on the disks that are accessed.

HBase write

In the case of writing data, clients don't interact directly with the underlying HFiles during writes, but it goes to two places: 1) the write-ahead log (WAL), HLog, and 2) the in-memory write buffer, MemStore. The WAL is a file on the underlying file system and a write isn't considered successful until the new WAL entry is successfully written. The MemStore is a write buffer where HBase accumulates data in memory before a permanent write, and the contents of MemStore are flushed to disk to form an HFile when the MemStore fills up. If HBase goes down, the data that has not yet been flushed from the MemStore to the HFile can be recovered by replaying the WAL, which is taken care of by the HBase framework.

Test drive

Let's start coding HBase. To administer your HBase instance, launch HBase by running the command **hbase shell** in the prompt. Now, to create a table, populate the values and to display the table, use the commands below:

- **create 'table','colf'** – creates a new table identified by 'table' and column family identified by 'colf'
- **put 'table','row','colf:a','value1'** – inserts a new record into the table, with the row identified by 'row'
- **scan 'table'** – returns the data stored in table
- **get 'table','row1'** – returns the records matching the row identifier provided in the table

Connection through Java program

Let's write a program to import the data from a file into a HBase table, with the name of the table being passed as the argument. In the Eclipse IDE, create a new project and write

a class named `readFromFile`. Add all the Hadoop jars into the build path of the project, ensuring that these six jars...

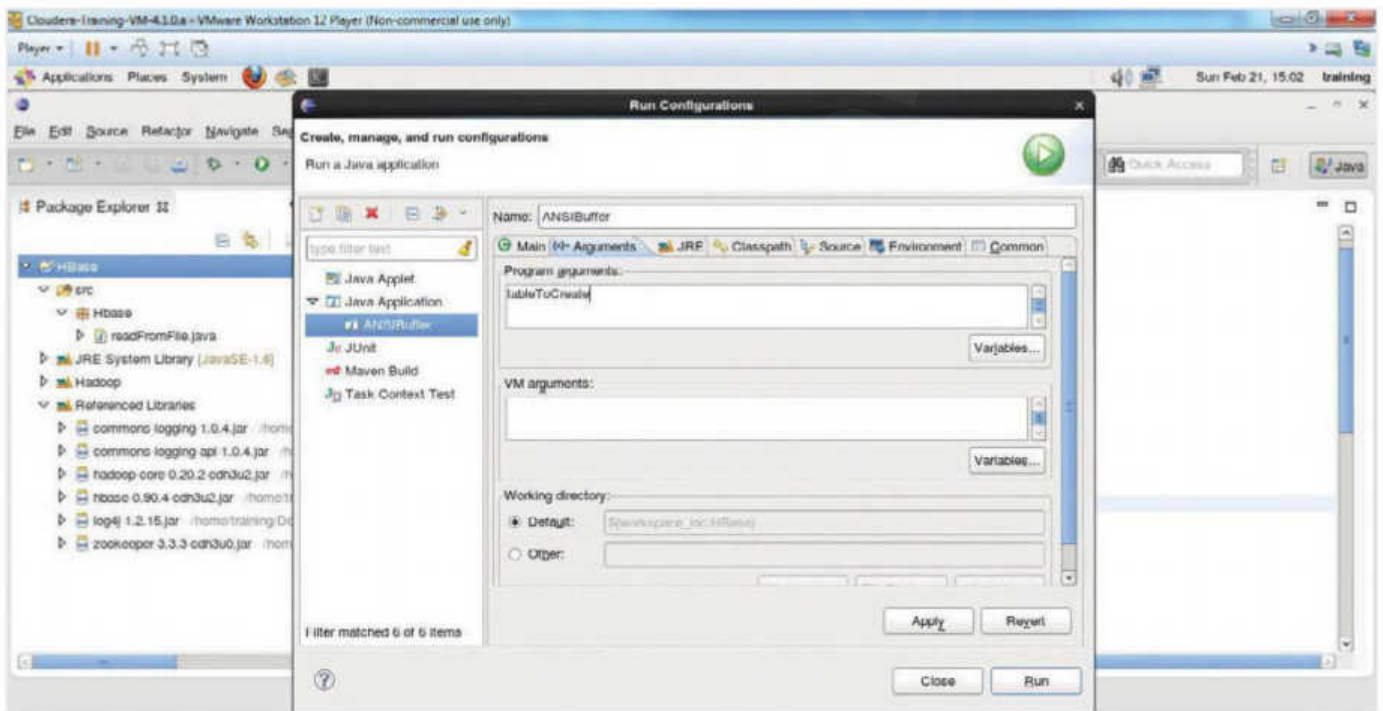
- 1) commons-logging-1.0.4.jar
- 2) commons-logging-api-1.0.4.jar
- 3) hadoop-core-0.20.2-cdh3u2.jar
- 4) hbase-0.90.4-cdh3u2.jar
- 5) log4j-1.2.15.jar
- 6) zookeeper-3.3.3-cdh3u0.jar

... are added as external jars in the build path.

Now, in the run configuration, pass the name of the table to be created as an argument; "tableToCreate" is the name of the table in this example.

```
package HBase;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.client.HBaseAdmin;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.util.Bytes;
```

```
public class readFromFile {
    public static void main(String[] args) throws
    IOException{
        if(args.length==1)
        {
            Configuration conf = HBaseConfiguration.
            create(new Configuration());
            HBaseAdmin hba = new HBaseAdmin(conf);
            if(!hba.tableExists(args[0])){
                HTableDescriptor ht = new HTableDescriptor
                (args[0]);
                ht.addFamily(new HColumnDescriptor("sample"));
                ht.addFamily(new HColumnDescriptor("region"));
                ht.addFamily(new HColumnDescriptor("time"));
                ht.addFamily(new HColumnDescriptor("product"));
                ht.addFamily(new HColumnDescriptor("sale"));
                ht.addFamily(new HColumnDescriptor("profit"));
                hba.createTable(ht);
                System.out.println("New Table Created");
                HTable table = new HTable(conf,args[0]);
                File f = new File("/home/training/Desktop/
                data");
                BufferedReader br = new BufferedReader(new
                FileReader(f));
                String line = br.readLine();
                int i =1;
                String rowname="row";
                while(line!=null && line.length()!=0){
                    System.out.println("Ok till here");
                    StringTokenizer tokens = new
```



```
StringTokenizer(line,",");
    rowname = "row"+i;
    Put p = new Put(Bytes.toBytes(rowname));
    p.add(Bytes.toBytes("sample"),Bytes.
toBytes("sampleNo."),Bytes.toBytes(Integer.
parseInt(tokens.nextToken())));
    p.add(Bytes.toBytes("region"),Bytes.
toBytes("country"),Bytes.toBytes(tokens.nextToken()));
    p.add(Bytes.toBytes("region"),Bytes.
toBytes("state"),Bytes.toBytes(tokens.nextToken()));
    p.add(Bytes.toBytes("region"),Bytes.
toBytes("city"),Bytes.toBytes(tokens.nextToken()));
    p.add(Bytes.toBytes("time"),Bytes.
toBytes("year"),Bytes.toBytes(Integer.parseInt(tokens.
nextToken())));
    p.add(Bytes.toBytes("time"),Bytes.
toBytes("month"),Bytes.toBytes(tokens.nextToken()));
    p.add(Bytes.toBytes("product"),Bytes.
toBytes("productNo."),Bytes.toBytes(tokens.
nextToken()));
    p.add(Bytes.toBytes("sale"),Bytes.
toBytes("quantity"),Bytes.toBytes(Integer.
parseInt(tokens.nextToken())));
    p.add(Bytes.toBytes("profit"),Bytes.toBytes
("earnings"),Bytes.toBytes(tokens.nextToken()));
    i++;
    table.put(p);
    line = br.readLine();
}
br.close();
table.close();
}
else
```

```
System.out.println("Table Already exists.
Please enter another table name");
}
else
System.out.println("Please Enter the table name
through command line");
}
```

Type scan 'tabletoCreate' into the shell to see the output.

```
###Data_File.png###
###Program.png###
###tableToCreate.png###
###Output.png###
```

RDBMS versus HBase

An RDBMS is a database that is of fixed schema. It is a row-oriented database with ACID properties and a sophisticated SQL query engine. In relational databases, the emphasis is on strong consistency, referential integrity, abstraction from the physical layer, and performing complex queries involving inner and outer joins, count, sum, sort, group by, etc.

On the other hand, HBase picks where Hadoop left off by providing random read/write on top of HDFS. It focuses on scalability, and is able to be horizontally partitioned and replicated across thousand of commodity nodes. For the applications working on small to medium datasets there is nothing better than an open source RDBMS like PostgreSQL, but when the data set is very large, RDBMS is too slow. Enter HBase, which does not have the issue of indexing, automatically portions data into regions, scales up linearly, and uses commodity hardware. ■

Above This is our simple Java program to read data from our HBase set

Refine your MonoGame shaders to boost performance

**Tam Hanna**

always loves a mathematical challenge.

As dimensioning electronic circuits get boring after some time, shader programming provides a neat alternative

Resources

MonoGame
monogame.net

Shaders can be used for much more than just for simple decorative tasks

Shaders let you run code on a massively parallel computer system: even the most primitive of GPUs bring along at least sixteen cores, along with a substantial amount of really fast RAM. In fact, for most computers, the GPU provides significantly more raw computational power than the CPU. Transferring processing workload to the GPU can lead to significant performance improvements.

In the last few years, shader use has been expanded from traditional rendering. Textures play a key role in such applications: they provide the only truly efficient way to pass data into and out of the GPU.

At the end of the previous article, we presented a little coding challenge to the willing reader. Applying post-processing effects to the results of a rendering process requires that said results lie in a texture – achieving that requires a technique called indirect rendering.

Indirect rendering

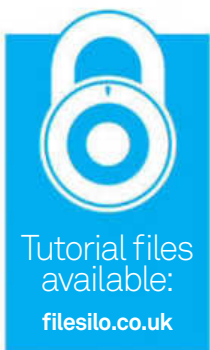
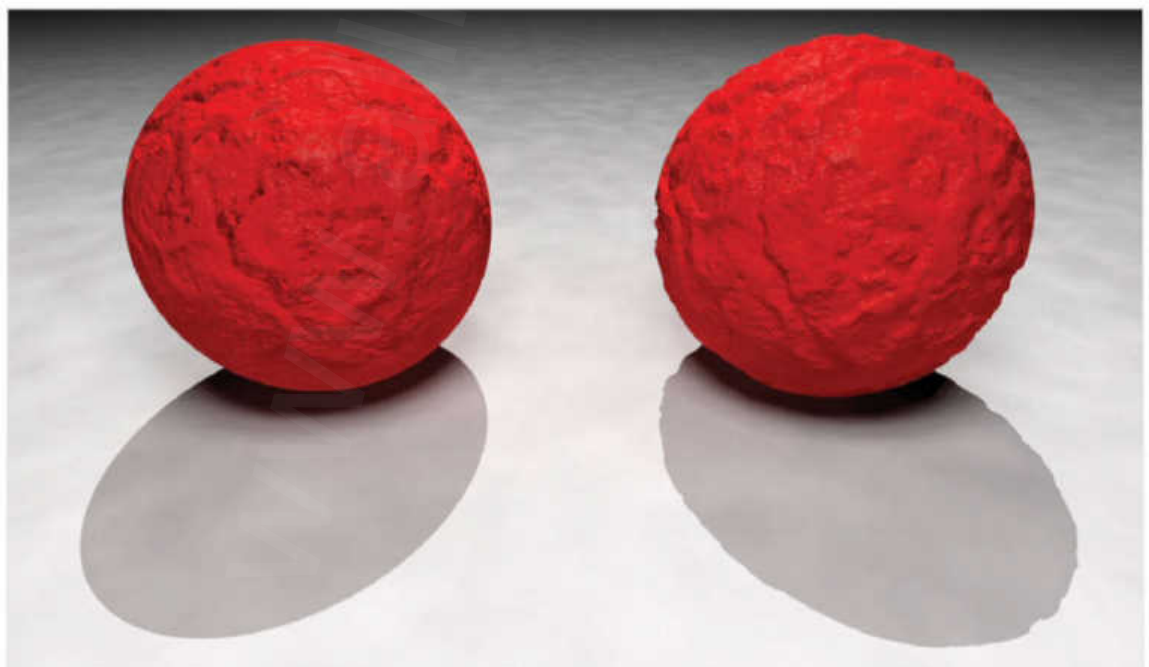
The cube we've worked with in the last couple of articles rendered directly into the display memory. As GPUs are not able to run shaders on the contents of this area, we must render the scene into a texture. This can be accomplished

by creating an instance of `RenderTarget2D`, which must be initialised in the `Initialize` method:

```
public class Game1 : Game
{
    . . .
    RenderTarget2D myRenderTarget;

    protected override void Initialize ()
    {
        . . .
        myRenderTarget = new RenderTarget2D(
            GraphicsDevice,
            GraphicsDevice.PresentationParameters.
            BackBufferWidth,
            GraphicsDevice.PresentationParameters.
            BackBufferHeight,
            false,
            GraphicsDevice.PresentationParameters.
            BackBufferFormat,
            DepthFormat.Depth24);
    }
}
```

Right The isosurface on the right models a sphere with a bumpy texture, whereas the bump-mapped sphere on the left still has a perfectly spherical outline



In the next step, `Draw()` needs to be modified so that the main rendering is redirected into the newly-created `RenderTarget`. This is accomplished by invoking the `SetRenderTarget` method of our `GraphicsDevice`:

```
protected override void Draw (GameTime gameTime)
{
    graphics.GraphicsDevice.Clear (Color.CornflowerBlue);

    //Start to Target
    GraphicsDevice.SetRenderTarget(myRenderTarget);
    GraphicsDevice.Clear(Color.CornflowerBlue);

    GraphicsDevice.DepthStencilState = new
    DepthStencilState() { DepthBufferEnable = true };
}
```

In the next step, the scene is rendered normally as if it were being drawn directly onto the screen:

```
//Render scene
myEffect.Parameters["World"].SetValue( Matrix.Create
Translation (0, 0, 0) * Matrix.CreateRotationX (0.2f));
myEffect.Parameters["View"].SetValue(Matrix.Create
LookAt(new Vector3(10,5,15), Vector3.Zero, Vector3.Up));
myEffect.Parameters["Projection"].SetValue(Matrix.
CreatePerspectiveFieldOfView(MathHelper.PiOver4,
graphics.GraphicsDevice.Viewport.AspectRatio, 1.0f,
1000.0f));
myEffect.Parameters["ModelTexture"].
SetValue(myTexture);

foreach (EffectPass pass in myEffect.
CurrentTechnique.Passes)
{
    pass.Apply ();
    graphics.GraphicsDevice.DrawUserPrimitives
    (PrimitiveType.TriangleList, myCube.myBones, 0, 12);
}
```

Once the rendering of the scene is complete, we proceed to discarding the render target. From that moment onwards, all rendering commands will be directed into the display memory once again:

```
GraphicsDevice.SetRenderTarget(null);

//Actual draw pass
myPassoverEffect.Parameters ["SourceTexture"].
SetValue (myRenderTarget);
myPassoverEffect.Parameters ["PassoverTexture"].
SetValue (myPassoverTexture);
myPassoverEffect.Parameters ["Threshold"].SetValue
(myPassoverStep);
```

Once the shader is set up, a `spriteBatch` is used to draw the contents of a screen-filling texture into the display memory:

```
spriteBatch.Begin (SpriteSortMode.Immediate,
BlendState.AlphaBlend,
SamplerState.LinearClamp, DepthStencilState.
```

```
Default,
RasterizerState.CullNone,myPassoverEffect);

spriteBatch.Draw(myPassoverTexture, new
Rectangle(0, 0, GraphicsDevice.PresentationParameters.
BackBufferWidth, GraphicsDevice.PresentationParameters.
BackBufferHeight), Color.White);

spriteBatch.End();

base.Draw (gameTime);
}
```

Casual observers will be perplexed by the use of `myPassoverTexture` in the `Draw()` call. This, however, is correct: as `spriteBatch` handles the entire drawing process via the `myPassoverEffect` shader, the texture passed into the `spriteBatch` is of no concern. You could easily pass in a pink or green one – our shader does not touch the values stored inside the `spriteBatch`.

Finally, `Update()` needs to be modified in order to change the value of the `passoverStep` variable in an acceptable speed:

```
protected override void Update (GameTime gameTime)
{
    . . .

    myPassoverStep += myPassoverDirection;
    if (myPassoverStep >= 1) {
        myPassoverDirection = -0.01f;
        myPassoverStep += myPassoverDirection;
    } else if (myPassoverStep <= 0) {
        myPassoverDirection = 0.01f;
        myPassoverStep += myPassoverDirection;
    }

    base.Update (gameTime);
}
```

Geometry, processed

Adding extra planes makes models look more realistic: the more triangles added, the closer the approximation between real and modelled shape. Sadly, getting too creative in 3DS Max is not a realistic option. Increasing the amount of vertices significantly increases the computational demands: while pure rendering grows with $O(n)$, collision detection algorithms can, in theory, grow with $O(n^2)$.

This situation provides an opportunity for a small reflection on life in general: if your goals can not be achieved, stepping back and taking a look at the intended benefits provides an enlightening new perspective. In the case of geometry, developers usually seek to increase the quality of the visual display. Small physical inaccuracies are insignificant in the large scheme of things – Geoffrey Dallimore's classic diagram (see the image on the opposite page) illustrates the point.

Bump mapping is easily explained: shaders compute the colour of the pixels shown on the screen. If developers modify the values used for the computation of the geometry, the resulting object's look can be modified to add extra geometry

Source does not exist!

Sometimes, the content pipeline throws 'The source file [name] does not exist!' errors. This is usually caused by mistakes in the Blender model: tell your artist to just unwrap the models – at the time of writing, adding textures in the UV view does nothing but confuse the content pipeline.

■ No grey heightmap?

Some (more academic) tutorials start out by demoing bump mapping with a simple greyscale heightmap which moves elements up and down in the direction of the normal. Our method provides more flexibility – most practical examples use multicoloured bitmaps for precisely that reason.

details without having to add actual vertices. This extra geometric information usually gets passed in as a texture called normal map. Each pixel contains information about a transformation to be applied to the point at hand – similarities to classic heightmaps are not purely coincidental.

Return to normals

Our 'classic' lighting shaders required the presence of normals information. As our first forays into texturing simply sampled colours from the bitmap, normals information was sacrificed. In theory, undoing this is not particularly difficult: MonoGame provides the **VertexPositionNormalTexture** class which can be used instead of **VertexPositionTexture**.

Sadly, we also need a bit of information called the Binormal – computing it at runtime is a very difficult task. Due to that, we switch to loading a prepared model that we have created – when working with a 3D editor such as Blender, exporting Binormal information usually takes place automatically. MonoGame's pipeline can also generate them for you if you set the Generate Tangent Frames field to true in MonoGame 3.5. Sadly, deploying this is not nearly as easy as it sounds. Start out by updating your Windows workstation with the installer file found at <http://www.monogame.net/downloads> in the development branch section, and also install the corresponding one on your Linux PC. Even though the installer promises to update MonoDevelop, this is not the case.

Next, you'll need to head across to <http://addins.monodevelop.com/Project/Index/47>. Here, you can download the latest version of the MonoDevelop plug-in by hand. Select the one suited to your MonoDevelop version, and open the Add-In Manager via the Tools menu. Next, delete the existing MonoGame plug-in and replace it with the one downloaded from the website.

Projects can not be upgraded as the new version of the plug-in uses a different toolchain: its MonoGame distribution no longer comes from a NuGet package, but is obtained from the GAC. Create a new project using the template MonoGame > App > MonoGame Cross Platform, then copy content and code files to the new project.

Finally, the actual bumpmap must be created. It is but a texture which can be converted using the Windows-only xNormal utility available at www.xnormal.net/Downloads.aspx. For the following steps, simply use the one provided in the example accompanying this story.

Down to maths

Performing bump mapping requires us to recompute the diffuse light information at every pixel. As our texturing shader currently does not consider realistic lighting at all, we must back-track to the shaders used in previous issues. The first change involves the adding of a batch of new input and output fields – normal, binormal and tangent information must be transferred:

```
struct VertexShaderInput
{
    float4 Position : SV_POSITION;
    float4 Normal: NORMAL0;
    float4 Tangent: TANGENT0;
    float4 Binormal: BINORMAL0;
```

```
    float2 TextureCoordinate : TEXCOORD0;
};
struct VertexShaderOutput
{
    float4 Position : SV_POSITION;
    float4 Normal: COLOR0;
    float3 Tangent: TEXCOORD3;
    float3 Binormal: TEXCOORD2;
    float2 TextureCoordinate : TEXCOORD0;
};
```

VertexShaderOutput contains a nice little trap. It allows only the intrinsics **COLOR** and **TEXCOORD** – if you use another intrinsic, the compiler will accept the file but it will fail when being run in MonoGame (can not compile).

Our vertex shader now transforms the data fields into a form more palatable to the pixel shader. The **WorldInverseTranspose** matrix is used for efficiency reasons – scaling with the normal World matrix leads to errors when elements are scaled axonometrically:

```
VertexShaderOutput VertexShaderFunction(VertexShaderInput input)
{
    VertexShaderOutput output;

    float4 worldPosition = mul(input.Position, World);
    float4 viewPosition = mul(worldPosition, View);
    output.Position = mul(viewPosition, Projection);

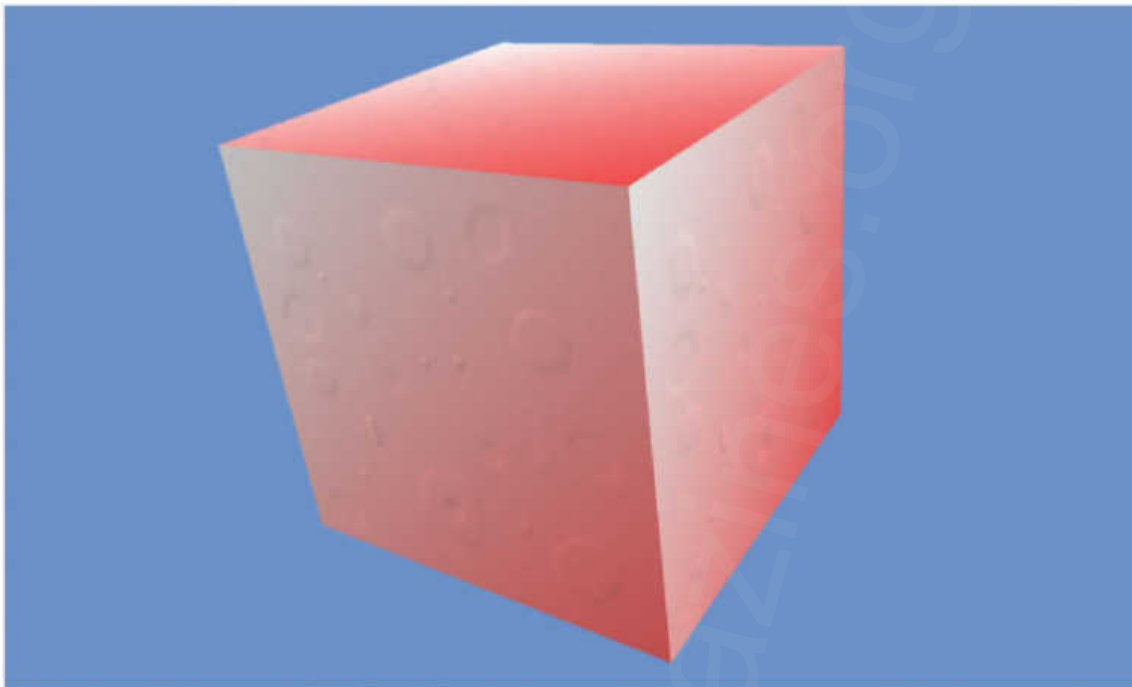
    output.Normal = normalize(mul(input.Normal, WorldInverseTranspose));
    output.Tangent = normalize(mul(input.Tangent, WorldInverseTranspose));
    output.Binormal = normalize(mul(input.Binormal, WorldInverseTranspose));
    output.TextureCoordinate = input.TextureCoordinate;
    return output;
}
```

The actual intelligence of the bump mapping process is found in **PixelShaderFunction**, which has the following corpus:

```
float4 PixelShaderFunction(VertexShaderOutput input) : SV_TARGET0
{
    float3 normalizer = float3(0.5, 0.5, 0.5);
    float3 bump = tex2D(bumpSampler, input.TextureCoordinate) - normalizer;
    float3 bumpNormal = input.Normal + (bump.x * input.Tangent + bump.y * input.Binormal);
    bumpNormal = normalize(bumpNormal);
```

Bump mapping does its magic by modifying the normal value used for the lighting computation. The normal map provides information about how far the virtual point must be moved in the direction of the tangent and the binormal, thereby permitting the creation of a "virtual object point".

The rest of the method is, by and large, normal shading code implementing a diffuse lighting process:



Left Here's our bump-mapped example, with raised and inset circles

■ Got the blues?

Careful observers of the pixel shader will note that pixels will be set to a light shade of blue if all three members of the tangent are zero. This is a helpful aid when debugging: should one of your model have a serious case of "feeling blue", checking its contents is highly recommended.

```
float diffuseIntensity = dot( bumpNormal,
    normalize(DiffuseLightDirection));
```

```
if (diffuseIntensity < 0)
    diffuseIntensity = 0;
```

```
float4 textureColor = tex2D(textureSampler, input.
    TextureCoordinate);
textureColor = textureColor *diffuseIntensity;
textureColor.a = 1;
```

```
if (input.Tangent.x != 0 || input.Tangent.y != 0 ||
    input.Tangent.z != 0)
```

```
    return saturate(textureColor + float4(1, 1, 1,
    1)*0.3);
else
    return float4(0, 1, 1, 1);
```

```
}
```

OnDraw must populate the various fields of the shader using the methods discussed in the previous issues:

```
foreach (ModelMesh mesh in myModel.myModel.Meshes)
{
    foreach (ModelMeshPart part in mesh.MeshParts)
    {
        //myEffect.Parameters["DiffuseColor"].
        SetValue(new Vector4(1,1,1,1));
        myEffect.Parameters["DiffuseLightDirection"].
        SetValue(new Vector3(1,1,1));
        myEffect.Parameters["World"].SetValue( Matrix.
        CreateTranslation (0, 0, 0) * Matrix.CreateRotationX
```

```
(0.2f) * Matrix.CreateScale(4)* mesh.ParentBone.
    Transform);
```

```
Matrix worldInverseTransposeMatrix = Matrix.
    Transpose(Matrix.Invert(Matrix.CreateTranslation (0, 0, 0)
    * Matrix.CreateRotationX (0.2f) * Matrix.CreateScale(4) *
    mesh.ParentBone.Transform));
```

```
myEffect.Parameters["WorldInverseTranspose"].
    SetValue(worldInverseTransposeMatrix);
myEffect.Parameters["View"].SetValue(Matrix.
    CreateLookAt(new Vector3(10,5,15), Vector3.Zero,
    Vector3.Up));
```

```
myEffect.Parameters["Projection"].
    SetValue(Matrix.CreatePerspectiveFieldOfView(MathHelper.
    PiOver4, graphics.GraphicsDevice.Viewport.AspectRatio,
    1.0f, 1000.0f));
```

```
myEffect.Parameters["ModelTexture"].
    SetValue(myTexture);
```

```
myEffect.Parameters["MapTexture"].
```

```
SetValue(myBumpyTexture);
```

```
part.Effect = myEffect;
```

```
}
```

```
mesh.Draw();
```

```
}
```

Finally, the creation of the world matrix is modified so that the cube appears larger:

```
myEffect.Parameters["World"].SetValue( Matrix.
    CreateTranslation (0, 0, 0) * Matrix.CreateRotationX
    (0.2f) * Matrix.CreateScale(4));
```

With that, it's time to run the program once again. It should yield results similar to the ones shown in the image above. ■

Set up your own DNS proxy server



Kunal Deo

is a long time contributor for the Linux User and Developer Magazine. Kunal has more than 11 years of experience in Linux and open source development.

Use dnsmasq and haproxy to redirect your traffic through a virtual IP address

Using VPN software enables you to have a completely different public IP address and region, which matches with the VPN server. This technique works but it does have significant disadvantages: your internet connection is now likely to be slow because everything is now going through the VPN server. It is also difficult to deploy VPN, as the client may need to support the VPN protocol used, and some devices do not support VPN in the same way as things like Android TV, the PS4, the Xbox One, etc. The solution to all of these problems is a DNS proxy server. It only redirects selected traffic so it does not affect the rest of the internet connection, and its deployment is also as easy as changing the DNS server settings.

Our DNS proxy server solution is spread across two servers. Firstly, the local server will provide three primary functions:

DNS entry for, say, [wikipedia.com](https://www.wikipedia.com) to one of your virtual IP addresses instead of the real IP address.

- Traffic Forwarder: Using tables we will re-route the traffic coming to each of the virtual IP addresses through to the remote server.

The remote server is located in the region of your choice. For example, to access US-specific content, it needs to be located in the US. The remote server's primary function is to route the incoming traffic transparently to the original server.

01 Install Dnsmasq server (local server)

Dnsmasq is a lightweight DNS server that works with minimal configuration. We will use dnsmasq to discover and override the DNS entries for the services that we want to access via a remote server. Install dnsmasq using:

```
$ sudo apt-get install dnsmasq
```

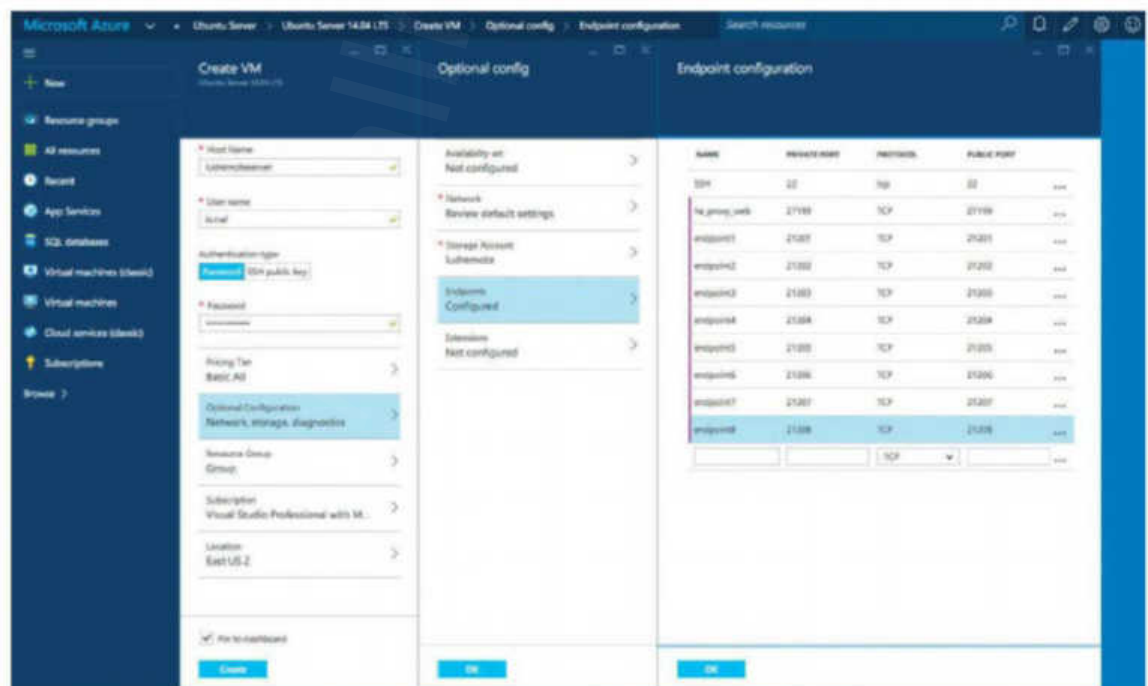
In `/etc/dnsmasq.conf`, enable the query log:

Resources

Dnsmasq
bit.ly/1XAorA4

Haproxy
haproxy.org

- Virtual IP Address Provider: Depending upon the number of services that you want to access, you will need to set up that many IP addresses.
- DNS Server: The DNS server will override DNS entries for the services you want to access. For example, it will override the



Right You'll need to set up a VM server in the region you want to route data through

```
log-queries
log-facility=/var/log/dnsmasq.log
```

Restart the dnsmasq service using:

```
$ sudo /etc/init.d/dnsmasq restart
```

02 Discover the DNS names

The first step of unlocking a service from geo-blocked restrictions is to find out which DNS names you need to route through remote servers. We have already enabled the dnsmasq's query logs – this allows us to see all the DNS requests coming its way. Open the dnsmasq streaming log using the command:

```
$ sudo tail -f /var/log/dnsmasq.log
```

Go to the device from which you want to access the service. Set up the DNS server IP address to the server you have installed. Now open the app or the website for the service you need access to. You will see an error message related to region block from the service. While this happens take a note of the streaming log for the addresses related to the service you were trying to access. If you are planning to access the services from other devices as well, you should repeat this process again to capture the device-specific addresses required by the service.

For example, if you are trying to access Facebook from your laptop you may notice following address related to Facebook service in the dnsmasq log:

```
www.facebook.com
pixel.facebook.com
```

If you are using the Facebook app on your iOS or Android device, you may see:

```
api.facebook.com
graph.facebook.com
```

Similarly, if you are using the Facebook app on your other devices you may see different DNS names. This process sometimes involves trial and error. The reason why we have opted to view a streaming log instead of the static log is that we need to monitor the real-time DNS requests when we access the service. Sometimes, the DNS name that's requested may not even resemble the original service name, but most of the time it does.

03 Creating virtual network adapters

For each of the addresses we have discovered, we need to create a virtual network adapter. This will allow us to create a set of virtual IP addresses which we can use to override the IP addresses for the discovered DNS names. In the above example, we may want to set api.facebook.com to 192.168.1.101 instead of the real DNS entry.

Edit /etc/network/interfaces and add the following lines to create a virtual interface:

```
auto eth0:101
iface eth0:101 inet static
```

```
address 192.168.1.101
netmask 255.255.255.0
```

```
auto eth0:102
iface eth0:102 inet static
address 192.168.1.102
netmask 255.255.255.0
```

```
auto eth0:103
iface eth0:103 inet static
address 192.168.1.103
netmask 255.255.255.0
```

```
auto eth0:104
iface eth0:104 inet static
address 192.168.1.104
netmask 255.255.255.0
```

Enable these virtual network adapters:

```
$ sudo ifup eth0:101
$ sudo ifup eth0:102
$ sudo ifup eth0:103
$ sudo ifup eth0:104
```

Here the address has been adapted for the network of 192.168.1.xxx series. If your LAN is using a different network, please update the IP address accordingly. Similarly, you can create more network interfaces. For this tutorial, we will assume virtual interfaces with IP addresses 192.168.1.101, 192.168.1.102, 192.168.1.103 and 192.168.1.104.

Since these addresses are static IP addresses, we will need to make sure the DHCP server on your network is not assigning the same IP addresses. In this case, we may want to configure our router for a DHCP server range of 192.168.1.1 to 192.168.1.100. This is, of course, is anticipating the fact that we will have only 100 devices on the network.

04 Configuring the dnsmasq server

In this step we need to map the discovered DNS names to the virtual IP addresses. Edit /etc/dnsmasq.conf to add the following lines:

```
address=/www.facebook.com/192.168.1.101
address=/pixel.facebook.com/192.168.1.102
address=/api.facebook.com/192.168.1.103
address=/graph.facebook.com/192.168.1.104
```

Restart the dnsmasq server to apply the changes. Now you can verify these settings by pinging the various DNS names and making sure that we are getting virtual IP addresses, and that they are reachable.

05 Forwarding traffic

In this step we will forward the requests coming to our virtual IP addresses: HTTP port (80) and HTTPS port (443) to remote server. To do this we will use a technique called DNAT (Destination NAT). NAT itself stands for Network Address Translation. NAT allows you to transparently map one IP address range to another IP address. DNAT allows you

■ Use a Raspberry Pi

A DNS server should be running 24/7, otherwise the systems depending on it will have no internet access. To keep the energy footprint low and hardware cost low, the Raspberry Pi is an excellent choice of hardware for running the local server. You can use the Raspbian OS and simply follow this tutorial to set up the local server for very little cost.

to publish a service from an internal network to a publicly accessible network.

If you don't want to go too deep into the technicalities of NAT and DNAT, you only really need to understand this: we are allowing the traffic to be served by a remote server transparently. We will use iptables to create DNAT mappings, which is a flexible firewall utility for Linux computers. To create a DNAT mapping, the following commands is used:

```
$ sudo /sbin/iptables -t nat -A PREROUTING -p tcp --dport <Local Port Number> -d <Local IP Address> -j DNAT --to-destination <Remote Public IP Address>:<Remote port>
$ sudo /sbin/iptables -t nat -A POSTROUTING -p tcp --dport <Remote Port> -j MASQUERADE
```

You can also write these commands into a shell script file, run it once and then save it with `iptables-persistent` or `iptables-save`.

For this example, we will assume the remote server is in US with a public IP address of 216.58.197.46. In a file called `dnatrules.sh`, create the following lines:

```
/sbin/iptables -t nat -A PREROUTING -p tcp --dport 80 -d 192.168.1.101 -j DNAT --to-destination 216.58.197.46:21201
/sbin/iptables -t nat -A POSTROUTING -p tcp --dport 21201 -j MASQUERADE
/sbin/iptables -t nat -A PREROUTING -p tcp --dport 443 -d 192.168.1.101 -j DNAT --to-destination 216.58.197.46:21202
/sbin/iptables -t nat -A POSTROUTING -p tcp --dport 21202 -j MASQUERADE
```

Now copy and paste those four lines into a new stanza. Increment the end of the first IP address in Line 1 (i.e. yours) by one – e.g. 192.168.1.102. Use the same IP address in Line 3. Now, there are two ports in each stanza at the end of the second IP address – note that Lines 1 and 3 are different ports. Increment each of these by two – e.g. 216.58.197.46:21203 in Line 1 and 216.58.197.46:21204 in Line 3. Add another two stanzas in this way, taking you up to 192.168.1.104 and then ports 21207 and 21208. Save the file and then run it:

```
$ sudo sh dnatrules.sh
```

06 Creating the remote server

The remote server is the key piece of the setup. It will accept the connection from the local server and provide the traffic for the requested services. The easiest way to set up a remote server is to use one of the virtual private server providers or cloud providers. There are many service providers to choose from, like Microsoft Azure, Digital Ocean, Linode, etc. Digital Ocean is one of the cheapest service providers charging \$5 a month, but you can use any other cloud provider.

Most of the cloud providers include some sort of trial so you can get started for free. For this tutorial we will use Microsoft Azure; it has the most generous trial offer of £125 in Azure credits. You can follow the similar process if using a different cloud provider. For now, sign up for the trial: <https://azure.microsoft.com/en-in/free>. Head over to portal.azure.com

and log in with your new account. Click on 'Virtual machines (classic)', click '+', search for Ubuntu 14.04, select it, then select Deployment Model as Classic.

In the Create VM dialog, enter your Host Name for the server and the User Name and Password for the server. Click Pricing Tier, select View All, then select A0 Basic and click Select – this is the cheapest one and it isn't resource-hungry. Now click Configuration > Network and make a note of the domain name. Now head to Endpoints Configuration, so we can configure the ports that we want to expose on the internet. From the local and remote mappings table, we need to open eight ports, and for each one you need to enter the following:

```
NAME: endpoint1
PRIVATE PORT: 21201
PROTOCOL: TCP
PUBLIC PORT: 21201
```

Once you have covered all the ports, click OK. Now go to Location and select where the service is available. In the case of Facebook, it is always available in the US so you should select one of the US regions. After making all the selections, click Create to create the VM. Once the VM has been created and started, note down the IP address of the domain name you have just created and make sure it is updated in the local server configuration.

07 Installing Haproxy server

HAProxy or High Availability Proxy is a popular TCP/HTTP open source load balancer. It is used to distribute server workload across multiple servers so that the services work properly even when there is high demand. But we don't have to worry about all that. For our purposes, we will use Haproxy to provide us with the capability to provide the traffic for the services that are geo-blocked.

First we need to install the haproxy server. For this tutorial we will be using version 1.5 of the haproxy server. Ubuntu 14.04 comes with 1.4 version of haproxy server, so we will use a PPA repo to install haproxy 1.5:

```
$ sudo add-apt-repository ppa:vbernat/haproxy-1.5
$ sudo apt-get update
$ sudo apt-get install haproxy
```

There are a lot of options that need to be configured here, but you need to pay attention to the frontend and backend section.

```
frontend f_facebook-com_http
bind 0.0.0.0:21207
mode http
```

Frontend sets up the interface where our local server will be sending the request.

```
backend b_facebook-com_http
mode http
server www.facebook.com www.facebook.com:80
```

Backend sets up the actual server which will then serve up the request.

08 Configuring HAProxy Server

The haproxy configuration file is located in the /etc/haproxy folder. Edit /etc/haproxy/haproxy.cfg with the following contents:

```
global
  daemon
  maxconn 20000
  user haproxy
  group haproxy
  stats socket /var/run/haproxy.sock mode 0600
  level admin
  log 127.0.0.1 local0 debug
  pidfile /var/run/haproxy.pid
  spread-checks 5
```

```
defaults
  maxconn 19500
  log global
  mode http
  option httplog
  option abortonclose
  option http-server-close
  option persist
  timeout connect 20s
  timeout client 120s
  timeout server 120s
  timeout queue 120s
  timeout check 10s
  retries 3
```

```
listen stats
  bind 0.0.0.0:27199
  mode http
  stats enable
  stats realm Protected\ Area
  stats uri /
  #Change this set your own password in the web interface
  stats auth haproxy:mypassword
```

```
frontend f_catchall_http
  bind 0.0.0.0:21000
  mode http
  option httplog
  capture request header Host len 50
  capture request header User-Agent len 150
  default_backend b_deadend_http
```

```
backend b_catchall_http
  mode http
  option httplog
  option accept-invalid-http-response
```

```
frontend f_catchall_https
  bind 0.0.0.0:21200
  mode tcp
  option tcplog
  tcp-request inspect-delay 5s
  tcp-request content accept if {req_ssl_hello_type 1}
  default_backend b_deadend_https
```

```
backend b_catchall_https
  mode tcp
  option tcplog
```

Due to space constraints we have not given the complete configuration file, but it is very easy to add the missing entries. You just have to repeat the blocks with frontend f_facebook-www_http, backend b_facebook-www_http combinations, with the correct DNS names.

For example, for **api.facebook.com** we need to add the following entries:

```
frontend f_api-facebook-com_http
  bind 0.0.0.0:21203
  mode http
  option httplog
  capture request header Host len 50
  capture request header User-Agent len 150
  default_backend b_api-facebook-com_http
```

```
backend b_api-facebook-com_http
  mode http
  option httplog
  option accept-invalid-http-response
  server api.facebook.com api.facebook.com:80 check
  inter 60s fastinter 10s downinter 10s fall 1800
```

```
frontend f_api-facebook-com_https
  bind 0.0.0.0:21204
  mode tcp
  option tcplog
  default_backend b_api-facebook-com_https
```

```
backend b_api-facebook-com_https
  mode tcp
  option tcplog
  server api.facebook.com api.facebook.com:443 check
  inter 60s fastinter 10s downinter 10s fall 1800
```

You need to make sure that, while running through the haproxy configuration, the values are in sync with the local server and the remote server. You can use the mappings table to create the correct configuration. To apply the changes, restart the haproxy server:

```
$ sudo /etc/init.d/haproxy restart
```

Finally, we need to configure the packet forwarding on the remote server:

```
$ sudo sysctl -w net.ipv4.ip_forward=1
```

In /etc/sysctl.conf:

```
net.ipv4.ip_forward = 1
```

This completes the server configuration. If you have followed the tutorial correctly and have both the local and remote server started, you should now have the Facebook app working for the US region on both your PC and mobile apps. ■

Simplify team communication with Mattermost

**Nitish Tiwari**

is a software developer by profession and an open source enthusiast by heart. As well as writing for leading open source magazines, he helps firms set up and use open source software for their business needs

Resources

Mattermost
mattermost.org

We'll see how to install Mattermost and use it to enhance and simplify team communication

Communication is a very important aspect of team dynamics. The productivity of a team depends on the quality of communication that people in the team have with each other. There has to be a balanced approach where the team members, while sharing a great rapport, do not invade the personal space of others. While a lot of this depends on the individuals themselves, or the company culture, the tools that you use have a very important role to play too. In this tutorial, we'll take a look at Mattermost, a tool that can help you streamline and simplify your team's internal communication, while being hosted behind your own firewall. Not only this, it also integrates well with other software like Jenkins, GitLab, SVN, etc to give a unified interface to your team's status.

Mattermost is available for download under the MIT licence from the official website. Mattermost installation is a simple affair with a one-line install available for Docker containers. Even if you'd rather install Mattermost on a Linux flavour, it is as simple as installing the database and launching the web server.

01 Database installation

As a web-based software with server-client architecture, Mattermost needs a database and a web server to run. Let us first install the database for Mattermost. We have used the PostgreSQL database for demonstration purposes on a host system with Ubuntu 15.10. First of all you need to update your system with the latest security updates and patches. Run:

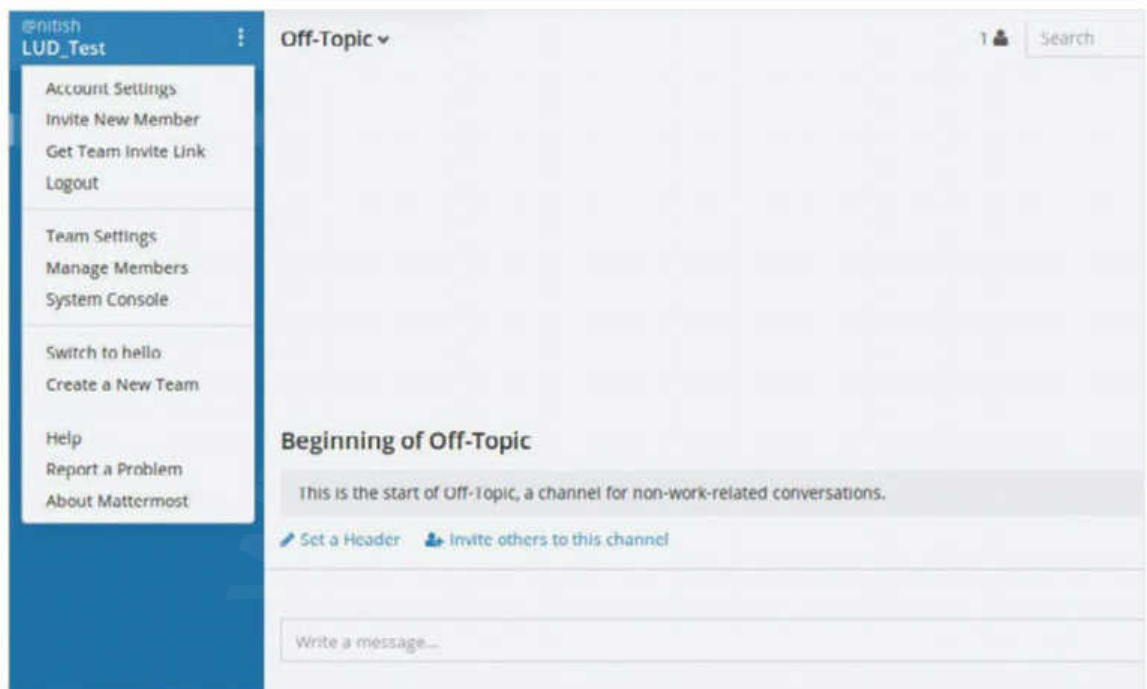
```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Now install the latest version of PostgreSQL on the system:

```
$ sudo apt-get install postgresql postgresql-contrib
```

PostgreSQL creates a user account called postgres. Log into that user account with:

```
$ sudo -i -u postgres
```



Right This is the Mattermost team home page showing one of the two default channels: Off-topic

```

root@ubuntu:~# sudo apt-get install postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libpq-dev libpq5 libssl-dev libssl1.0.2 libssl1.0.2-dev libssl1.0.2-gnutls
libssl1.0.2-openssl
The following extra packages will be installed:
libpq5 postgresql-client-9.4 postgresql-client-common postgresql-common postgresql
Suggested packages:
postgresql-doc postgresql-server-dev-9.4 postgresql-doc-9.4 libpq-dev
The following new packages will be installed:
libpq5 postgresql postgresql-client-9.4 postgresql-client-common postgresql-common
0 upgraded, 6 newly installed, 0 to remove and 7 not upgraded.
Need to get 4,308 kb of archives.
After this operation, 28.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

Next, go to the PostgreSQL prompt by typing:

```
$ psql
```

Let us now create a Mattermost database and a user account to manage the database:

```

# CREATE DATABASE mattermost;
# CREATE USER mmuser WITH PASSWORD 'mmuser_
password';
# GRANT ALL PRIVILEGES ON DATABASE mattermost
to mmuser;

```

With that sorted, we now need to test if we can log in to the Postgres account and access the Mattermost database we created. Type:

```
$ psql --host=localhost --dbname=mattermost
--username=mmuser --password
```

This should take you to the mattermost prompt, indicating that the login was successful.

02 Mattermost installation

We have the Database and the related account ready, let us now download and install Mattermost:

```
$ wget https://github.com/mattermost/platform/releases/download/v2.0.0/mattermost.tar.gz
```

This will download the tarball with Mattermost code. Unzip the tarball by typing:

```
$ tar -xvzf mattermost.tar.gz
```

This will unzip and place the contents in a folder called mattermost in the current directory. Now create a folder to save the data files:

```
$ sudo mkdir -p /mattermost/data
```

The next step is to edit the config.json file located inside /mattermost/config folder. Type:

```

$ cd /mattermost/config
$ sudo vi config.json

```

Then replace the line “DriverName”: “mysql” with “DriverName”: “postgres”. The next line that you need to find should be just below “DriverName”: “DataSource”: “mmuser:mostest@tcp(dockerhost:3306)/mattermost_test?charset=utf8mb4,utf8”. Replace it with “DataSource”: “postgres://mmuser:mmuser_password@localhost:5432/mattermost?sslmode=disable&connect_timeout=10”. This sets the database driver for the PostgreSQL database that we just installed and configures Mattermost to connect to the database as the mattermost user.

Let us now run the Mattermost server and see it in action. First, go to the folder /mattermost/bin:

```
$ cd /mattermost/bin
```

Then run the Mattermost server by typing:

```
$ ./platform
```

You should see a console log like “Server is listening on :8065”, indicating that the service is running. Open the browser and type <http://localhost:8065> to see the Mattermost welcome page. You can stop the server by pressing Ctrl+C.

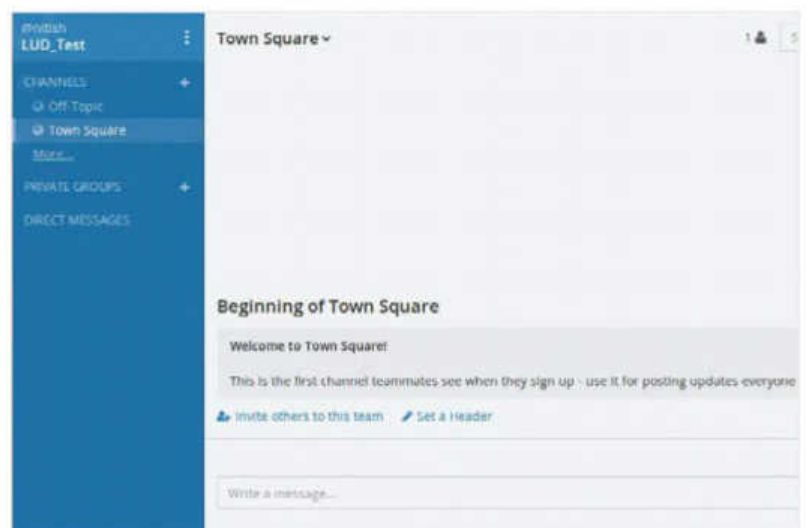
03 Getting started

Now that Mattermost is successfully installed and you can access the homepage on your browser, let us see how to get started. First, you’ll be asked to create a team by entering your email address. Then, enter the team name. Once you enter the team name, you’ll have to confirm the team URL that Mattermost suggests. Alternatively, you can change the URL as per your preferences and then confirm the URL. Then you’ll be asked to choose a username and password. You can use the id and password to login to this channel now onwards. After you log in, you’ll be taken to the Town Square channel of your team. Town Square is one of the two default public channels. Teams can use it to broadcast general information.

You can also add members to the team or set the team header, which is the description for the team from here.

Theme colours

The colours of the Mattermost user interface are customisable in Account Settings > Display > Theme. You can customise the colours yourself, choose from four standard themes designed by the Mattermost team, or import your theme colours from Slack. To make custom adjustments on the four standard theme colours, click a standard theme and then select Custom Theme to load the standard theme into the custom theme colour selectors.



■ Searching

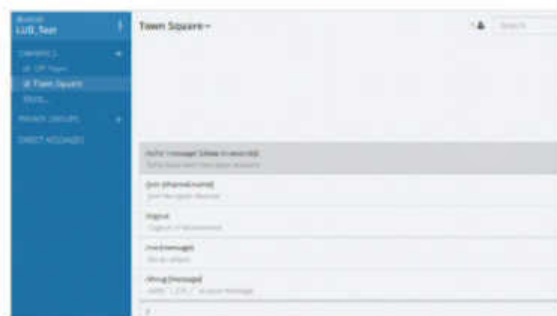
The search box in Mattermost returns non-case sensitive results from all of the channels where you're a member. You can also use search modifiers to narrow your search results. For example, Use from: to find posts from specific users and in: to find posts in specific channels. Also, if you'd like to view that post in the channel archive, just click Jump on the right of a search result.

04 Messaging basics

The bottom text bar lets you enter messages in a channel or one-to-one chats. To create a new line without sending the message you can press Shift+Enter simultaneously. You can mention teammates in your messages by typing "@username". It is interesting to note that the text area supports Markdown, so you can easily format your messages with just basic knowledge of Markdown syntax. Here is a primer:

Text you enter	Displayed as
bold	bold
<i>_italic_</i>	<i>italic</i>
[hyperlink](http://www.linuxuser.co.uk/)	hyperlink
'code'	code
~~strike through~~	~~strike through~~

Mattermost also has an in-built file previewer that lets you take a peek at attached files easily. Just click the attached file to enable the file preview. After a preview opens, you'll notice the Download and Get Public links at the bottom. While the Download option is self explanatory, Get Public enables you to share the files with people outside your Mattermost installation.

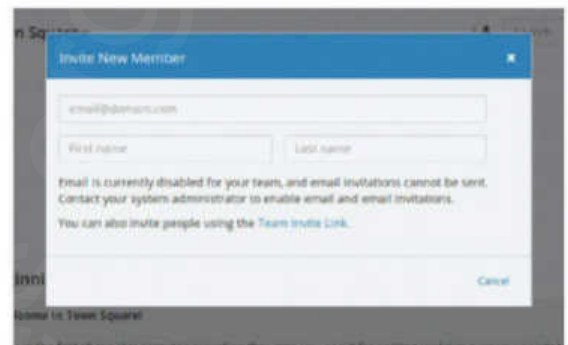


05 Executing commands

Mattermost lets you execute commands via the message text area. There are several built-in commands that can be used to perform basic tasks quickly – here are few of the built-in commands:

- **/echo message [delay in seconds]:** This command enables you to send a message as yourself after a delay of the specified number of seconds
- **/join channel-name:** Lets you join a channel directly
- **/logout:** Logs you out of Mattermost

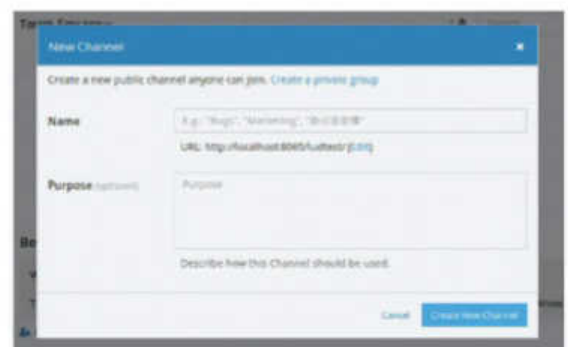
Mattermost also supports autocomplete, so you can see all the built-in commands by simply typing "/" in the message text area. It is also important to mention here that you can add custom commands to your Mattermost installation. These commands work on a basic HTTP request response mechanism and can be easily added to Mattermost. For example, if you want to add a command to communicate with an external application, you just need to handle the HTTP response from the external application and format it to properly display the results in the Mattermost channel.



06 Adding members

There are two ways to invite new members after team creation: sending direct invites and sharing the team invite link. Direct invites are emails sent from your team server directly to the invited member's email address. A link within the email directs the users to an account creation page. To send an invite, select "Invite New Member", from the Main Menu (three-dots icon) in the Mattermost header. Then, enter the new member's email address with their first and last name. Click "Add another" to send more direct invites from the same dialog.

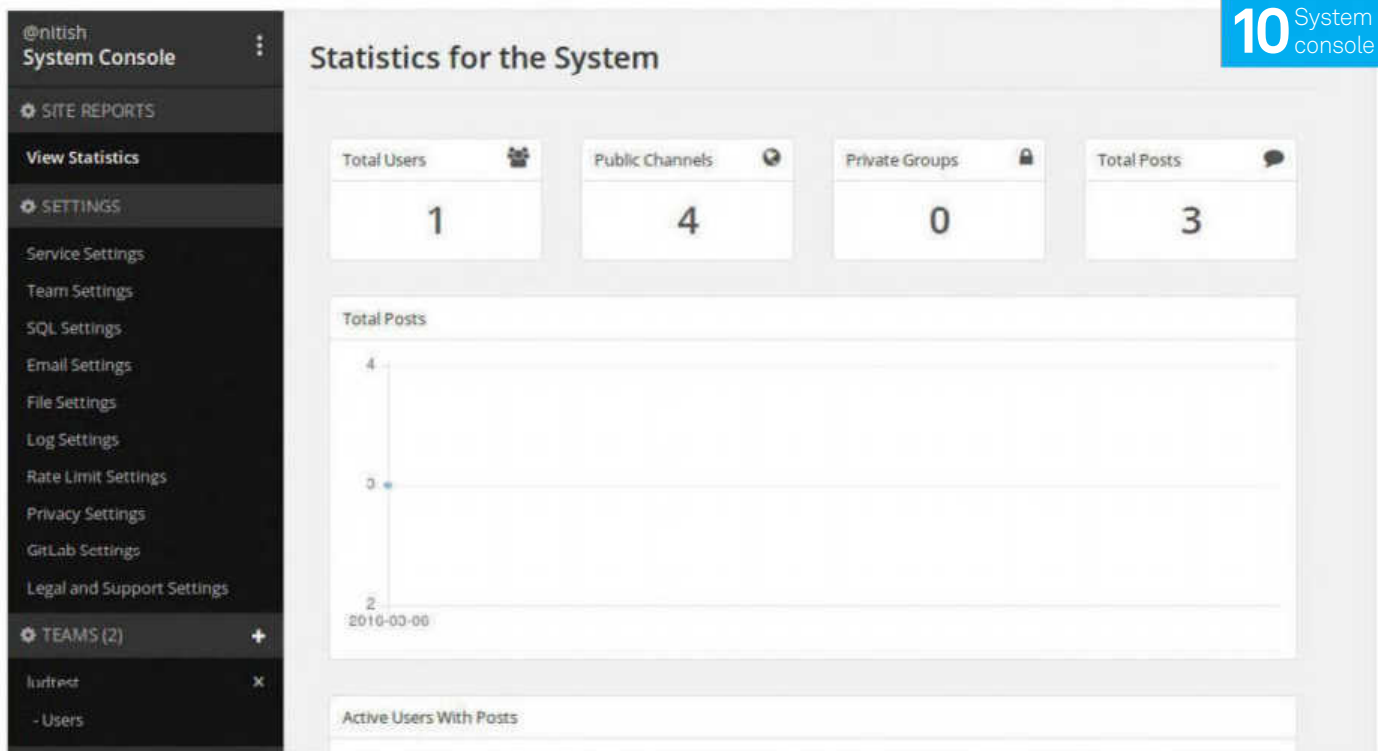
The team invite link is a unique URL that directs users to an account creation page for your team. Your team invite link can be used by any number of users and does not change unless it is re-generated. For example, the team invite link can be included in a company-wide email to invite all employees to join a Mattermost team. To create a link, click on the Main Menu icon in the Mattermost header again, this time selecting "Get Team Invite Link". This is the invite link, so just copy and share it with users you'd like to invite to the team.



07 Managing channels

To create a new channel or a private group, select the + symbol next to the corresponding header on the left-hand side. To start a direct message thread, click "More" at the bottom of the Direct Messages list to view a list of team members you can message. To join a channel, click "More" at the bottom of the Channels list to view a list of Public Channels you can join. To join a Private Group you need to be added by a member.

To leave a channel, click the channel name at the top of the centre pane to access the drop-down menu, then click "Leave Channel". Note that any team member who leaves a Private Group must be added back by a group member if they wish to rejoin. Also, team members will not receive mention notifications from channels of which they are not members.



10 System console

08 Organising conversations

There are three different ways to communicate in Mattermost: public channels, private channels and direct messages. Public channels are open to everyone on a team. New team members are automatically a part of two Public Channels: Town Square and Off-Topic, while private groups can be used to discuss potentially sensitive topics and are only visible to select team members. Any member of a private group can add additional members. Group members can choose to leave at any time, but only the group owner or team admin can remove other members.

Direct Messages are for conversations between two team members. These conversations are only visible to the team members involved. The online status indicator next to team members' names in the direct message list displays: online (active browser), away (no browser activity for five minutes), and offline (browser closed).

As you log on to Mattermost, you can view all of these streams of conversations on the left-hand menu bar, clearly segregated under appropriate headers.

09 Notification configuration

Unread messages in a channel are indicated by a bold channel name in the left-hand side bar to notify a new message. Clicking the channel name takes it back to normal, implying that the message has been read. You can manage the events that trigger the channel bolding by clicking Channel Menu > Notification Preferences > Mark Channel Unread.

Similarly, bold text and a mention jewel next to the channel name in the left sidebar indicate unread mentions in a channel. Clicking the channel name removes the bold style and mention jewel. You can also review your recent mentions by clicking

the @ symbol beside the search box at the top of the screen. There are email notifications available, too. Manage email notifications in Account Settings > Notifications > Email Notifications. To configure the email address where notifications are sent, go to Account Settings > General > Email. A notification sound plays for all activity that would fire a desktop notification. Notification sounds are available on Safari, Chrome and Edge. You can manage these via Account Settings >> Notifications >> Desktop Notification Sounds.

10 System console

As a system administrator you can configure several options using the system console. Note that all the settings done via the system console are stored in the /config/config.json file, so it is important to keep the config file in a safe and secure location to make sure no one gets to change it other than the admins. The first user added to a new Mattermost installation is assigned the system administrator role and can access the system console from the main menu of any team.

To access the system console, click on the three-dots icon on the top left corner and select "System Console". By default the console displays the system statistics, like total number of users, total public/private channels, etc. You'll also see a graphical representation of total number of posts and active users. All the other options available under system console can be seen on the left sidebar. For example, you can change the default port to listen for connections, the maximum number of login attempts, and several other developer-related settings here. If you want to add custom commands, you'll need to enable incoming and outgoing webhooks from here as well. ■

Above Usage stats for each of your teams are provided inside the console



Mihalis Tsoukalos

is a Unix administrator, a programmer (for Unix and iOS), a DBA and also a mathematician. He has been using Linux since 1993

Resources

Text editor
GCC compiler
yacc
flex
lex
bison
GNU make

Computer science Programming Languages

Put your newfound computer science skills to the test by writing a simple programming language from scratch

This tutorial will present the necessary knowledge and code that you need to know in order to create a programming language. The programming language that is going to be developed will emulate the functionality of a simple integer calculator; it might seem strange at first but a calculator is implemented using a simple programming language. You can better understand this if you realise that the “---1” string is not supported by any ordinary calculator, which means that it does not belong to the language of the calculator.

The presented example will be implemented in C with the help of some truly sophisticated UNIX command line tools. The main benefit you get from using language development tools is that they allow you to concentrate on what really matters, which is the actual design of the programming language, and give you the opportunity to experiment easily.

The theory

Developing a programming language is a difficult task, so make sure that you really understand the theory before continuing with the rest of the tutorial – despite the simplicity of the presented programming language, the

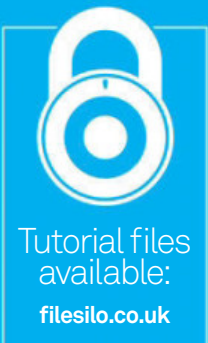
general principles are the same. If at any point you feel perplexed, we recommend returning to the theory and making your own notes on the subject.

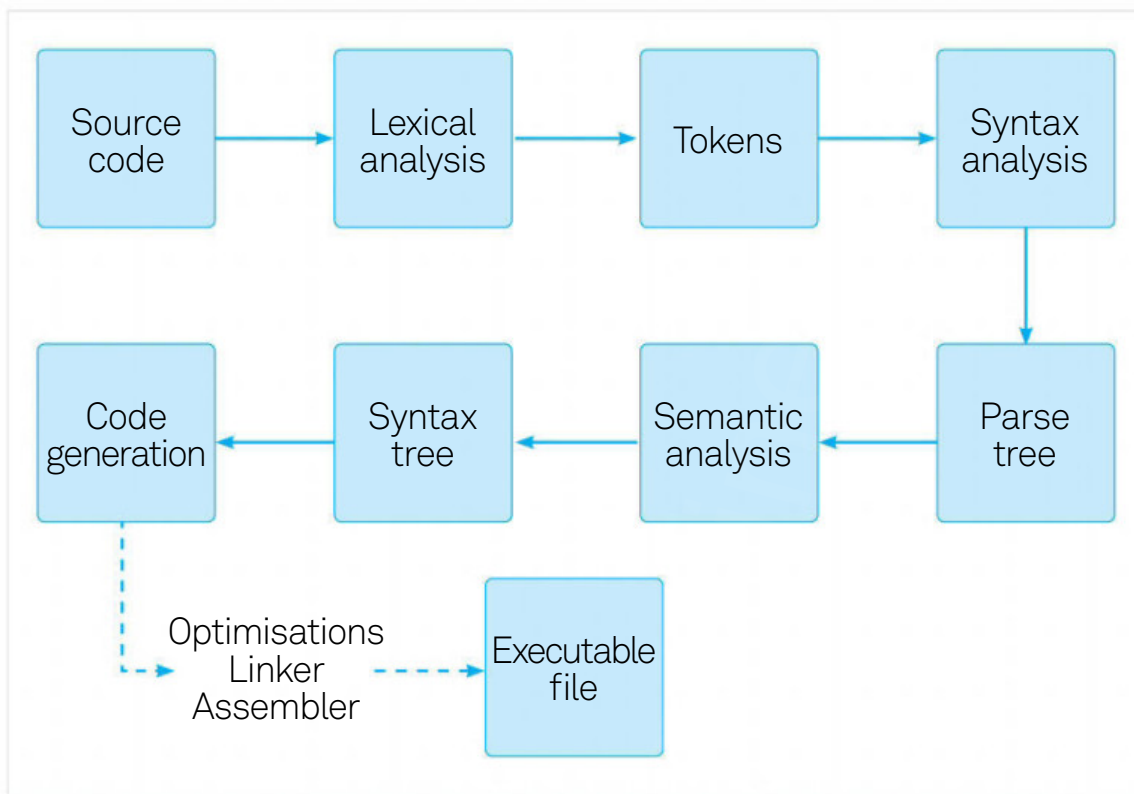
A dictionary is the set of the special characters and words of a programming language. Usually, what belongs to the dictionary cannot be used elsewhere. As an example, consider the word “int”, which in C has only one meaning unless it belongs to a string and cannot be used as a variable name.

A token is a word or a punctuation symbol. A grammar is a set of production rules for making strings in a formal language. A grammar does not describe the meaning of a string or what can be done with it, only its form.

In a context-free grammar F , if there is a production rule in the form $X \rightarrow Xb$, where X is a non-terminal and “a” is a string of terminals, it is called a left recursive production. Similarly, production rules of the form $X \rightarrow bX$ define a right recursive grammar.

A non-terminal symbol is a symbol that must be further expanded to create a sequence of terminal symbols. A compiler is a program that converts source code from one programming language into another computer language,





Left This is an overview of how a compiler works, from the code input to the file output

known as object code. Then, a linker creates an executable binary file from object code files and programming language libraries. Most compilers automatically call the linker so that you don't have to do it yourself.

An interpreter is a program that directly executes source code line by line. However, both compilers and interpreters have to do the same amount of work, so do not underestimate an interpreter!

How compilers and interpreters work

The diagram above shows how compilers work and the various phases of the compilation process. As you can understand, it is a relatively challenging activity where many elements must be taken care of before you get the executable file – think of it the next time you have a bug in one of your programs!

So, first the compiler gets the source file, which is a character stream, and processes it using the scanner, which does the lexical analysis of the source file. The scanner produces tokens – actually a stream of tokens – which are then used as the input to the parser that does the syntax analysis and produces the parse tree. Then, semantic analysis takes place and code generation begins. You now have a syntax tree or an equivalent of a syntax tree. The last steps are all about code generation, code optimisation and creating the executable file, where among other tools a linker is involved.

Put simply, tools such as lex and flex do the lexical analysis and split the input into tokens, whereas tools such as yacc and bison do the semantic processing using the tokens that flex and lex produce.

The lex utility

Mike Lesk developed the original version of lex around 1975 at the AT&T Bell Labs. The lex utility can help you make lexical analysers. Its main job is to break up an input stream into usable elements, which are called tokens. Lexical analysers are naïve tools that do not deal with the true meaning of the tokens – this is the job of a parser such as yacc or bison. The lexical analysis phase usually eliminates all blanks, because apart from helping you separate tokens they have no other use. As you will see, the lex utility has been completely replaced by flex.

The yacc utility

The name yacc is an acronym for Yet Another Compiler Compiler, which means that yacc is a utility for creating compilers for programming languages.

In computer science terminology, yacc is a parser generator. This means that yacc gets a grammar of a programming language as its input and generates the source code of a parser as its output. It is the job of the parser to determine the meaning of each token and whether a variable has a valid name or not and maybe reject it and generate error messages.

Strictly speaking, yacc is a LALR (look ahead, left-to-right) parser generator, which is a software tool that reads a BNF (Backus-Naur form) grammar and creates an LALR parser. The LALR parser is able to separate and analyse a text according to a set of production rules for left-to-right. The production rules are specified by a formal grammar. A BNF grammar is used for specifying the syntax of the programming language. Yacc has been entirely replaced by bison.

Regular grammars

A regular grammar is a set of production rules for strings in a formal language. The production rules describe how to create strings from the alphabet of the language that are valid according to the fixed syntax of the language. A grammar does not describe the meaning of a string or what can be done with it in whatever context – only its form. Each regular grammar describes a regular language. Therefore, you cannot define a programming language without a properly defined regular grammar.

The bison utility

Bison is the improved and open source version of yacc that is backwards compatible with yacc. Take a quick look at its man page to see what options are available for it.

In order for Bison to be able to parse a given language, the language must be described by a context-free grammar in a bison specification file. A bison specification file has four sections; the second, third and fourth sections are separated by lines having `%%`:

```
%{
C declarations
%}
```

```
Bison declarations
%%
Grammar rules
%%
```

Additional C code

In the “C declarations” part you can put `#include` C commands to get the declarations you want from a C header file as well as other C code. If you do not need such declarations, you can omit this part. If present, this part is copied at the beginning of the generated parser file. The “Bison declarations” segment contains declarations that among other things define terminal and nonterminal symbols as well as precedence for various operators. The “Grammar rules” part must contain at least one grammar rule. The “Additional C code” section is copied to the end of the generated parser file.

A bison grammar rule has the following format:

```
result: components...
;
```



Right Flex is a powerful utility – it's worth familiarising yourself with the available command line arguments

“The developed calculator implements the functionality of a basic calculator; it supports addition, subtraction, etc”

The result is a nonterminal symbol, whereas components are various terminal and nonterminal symbols defined somewhere in the specification file. Rules can be recursive, meaning that the result keyword can also appear on the components part of a rule.

If you execute bison using a file that only contains the two lines with `%%`, you will get the following error message:

```
$ bison myCalc.y
myCalc.y:6.1-2: syntax error, unexpected %%
```

A wrongly defined rule inside myCalc.y produces the following kinds of error message:

```
myCalc.y:6.1-6: syntax error, unexpected identifier
myCalc.y:4.9-17: symbol statement is used, but is
not defined as a token and has no rules
```

The flex utility

The flex utility is a fast lexical analyser generator. An alternative name for flex is scanner generator. Its input file is a set of regular expressions; an action written in C is associated with each regular expression.

You can see a small part of the flex man page to the left – you can find more information about flex if you type the following command:

```
$ info flex
```

The flex input file has three parts – the first and third parts are optional – and has the following structure:

```
FIRST PART
%%
SECOND PART
%%
THIRD PART
```

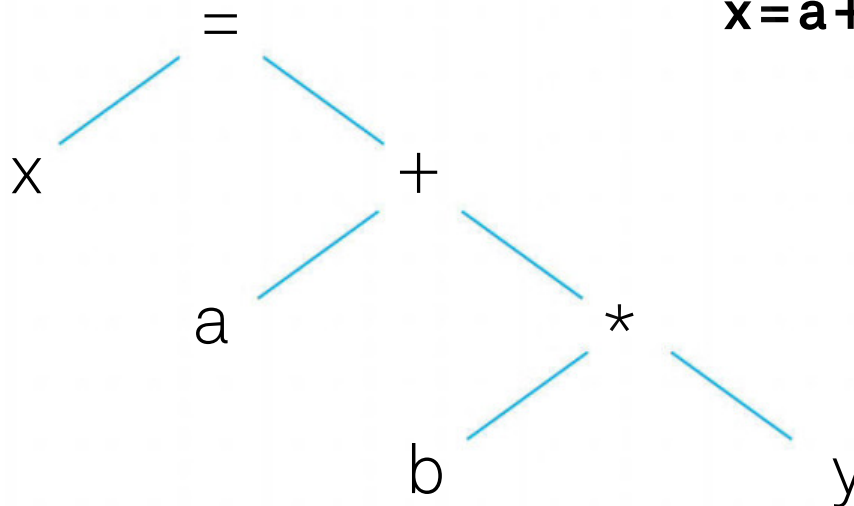
The second part has rules of the following form:

```
pattern    action
```

Please note that the order you put the rules is significant because only the first rule that is a match will be executed. Trying to process an empty file with flex produces the following error message:

```
$ flex myCalc.l
myCalc.l:1: premature EOF
```

Syntax tree



Expression
x=a+b*y

Left We can portray the BIDMAS rule for an algebraic expression using a binary tree structure

Test.l, also provided in your resources, is a small flex example that runs without the need for a bison file:

```
$ cat test.l
%%
"Hello LUD" printf("Hello Linux User &
Developer!\n");
%%
$ flex test.l
$ gcc lex.yy.c -ll
$ ./a.out
Hello LUD
Hello Linux User & Developer!
```

The `-ll` option tells the compiler to get the `main()` function from the flex library. The `test.l` file does nothing impressive but it generates an executable file that works and acts as an interpreter for the "language".

The interesting code inside `lex.yy.c` –generated by flex– is the following:

```
case 1:
YY_RULE_SETUP
#line 3 "test.l"
printf("Hello Linux User & Developer!\n");
YY_BREAK
case 2:
YY_RULE_SETUP
#line 4 "test.l"
ECHO;
YY_BREAK
```

The first rule says what to do when it finds the rule in the third line of `test.l`, whereas the second rule says that for any other

text it should act as an echo server that repeats the input. The generated `lex.yy.c` file is relatively big, considering that it only implements one rule:

```
$ wc lex.yy.c
1734 6157 44144 lex.yy.c
```

The required steps

When you execute flex and bison and make them process their input files, what you are going to get are files that contain C source code – this means that you will need to compile them to get the desired executable.

Your own code will need to call the `yyparse()` function that was created when bison processed its input file. The `yyparse()` function will then call `yylex()`, which was generated by flex using its scanner specification file.

The `yyparse()` function keeps calling `yylex()` in order to get all tokens from your input, which is a program in the programming language you want to create.

The single most important step is the definition of the grammar of the language. The diagram above illustrates the syntax tree of a simple expression to let you understand the internal representation of a small expression. Imagine how big and complex the syntax tree would be for a C program!

Supported functionality of the calculator

The developed calculator implements the functionality of a basic calculator: it supports the addition, subtraction, multiplication and division of integers.

Please note that when performing a division, the current version does permit division by zero. Additionally, the calculator only implements integer division, which means that the result of a division will always be an integer.

The "quit" keyword will end the calculator interpreter. Finally, the calculator will also exit when any character that

Further reading

The definitive guide to the theory of compiler construction is the book *Compilers: Principles, Techniques, and Tools* by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, also known as *The Dragon Book*. This is a difficult book with lots of theory but it contains tons of useful information you won't regret learning. You can find more about at dragonbook.stanford.edu/. Another truly useful book is *UNIX Programming Environment* by Kernighan and Pike.

```

-rw-r--r-- 1 mtsouk mtsouk 230 Feb 23 22:31 Makefile
-rw-r--r-- 1 mtsouk mtsouk 296 Feb 23 22:31 myCalc.l
-rw-r--r-- 1 mtsouk mtsouk 657 Feb 23 22:31 myCalc.y
drwxr-xr-x 2 mtsouk mtsouk 4096 Feb 23 19:34 test
code$ make
bison -d myCalc.y
flex myCalc.l
gcc -Wall -g lex.yy.c myCalc.tab.c -o myCalc
lex.yy.c:1104:17: warning: 'yyunput' defined but not used [-Wunused-function]
    static void yyunput (int c, register char * yy_bp )
                   ^
lex.yy.c:1145:16: warning: 'input' defined but not used [-Wunused-function]
    static int input (void)
                   ^
code$ ll
total 148
-rw-r--r-- 1 mtsouk mtsouk 44971 Feb 23 22:51 lex.yy.c
-rw-r--r-- 1 mtsouk mtsouk 230 Feb 23 22:31 Makefile
-rwxr-xr-x 1 mtsouk mtsouk 37216 Feb 23 22:51 myCalc
-rw-r--r-- 1 mtsouk mtsouk 296 Feb 23 22:31 myCalc.l
-rw-r--r-- 1 mtsouk mtsouk 44367 Feb 23 22:51 myCalc.tab.c
-rw-r--r-- 1 mtsouk mtsouk 2079 Feb 23 22:51 myCalc.tab.h
-rw-r--r-- 1 mtsouk mtsouk 657 Feb 23 22:31 myCalc.y

```

Right Here you can see our C and header files being generated by the build utilities

Bootstrap

'Bootstrapping' has many meanings – in a compiler context, it is the process of writing a compiler for a new programming language in the new language itself! You first write a compiler for the new programming language in another existing language. After you have a working compiler written in the other language, you create a compiler for the new programming language that its code is written using the new language.

is not supported is given as input. Should you wish to improve the current version, you can try to add variables, support floating point numbers, detect division by zero, add support for parenthesis and mathematical functions, etc.

The grammar for the calculator

The part of the bison file that contains the grammar of the language is very important because the grammar is the actual definition of the language. All other stages of compiler construction depend on the grammar used. Please note that different context-free grammars can generate the same context free language.

The simple grammar that will be used in this tutorial can be found on the third part of the bison file.

The code part

The parser that bison creates is in reality a finite state machine with a stack. A finite state machine is a state machine with a limited number of possible states. A state machine is any device that saves the status of something at a given time, and which can then operate according to its input in order to change the status and/or cause an action for any given modification.

The compiler or the interpreter produces a derivation tree that exhibits the structure that the grammar associated with the given string. The process of producing this tree is called parsing. This is how a compiler decides whether it has to deal with a valid program or not.

The code for the scanner specification file (myCalc.l), used by flex, is the following:

```

%{
#include <stdlib.h>
#include "myCalc.tab.h"
extern int yylval;

```

```

void yyerror(char *);
%}
%%
"quit"      {return quitCommand;}
[0-9]+      {yylval = atoi(yytext); return NUMBER;}
[ \t]       ;
[-+/*=\n]   {return *yytext;}
.           {yyerror("Invalid character!"); return 0;}

```

```

%%

```

```

int yywrap(void)
{
    return 1;
}

```

The line that begins with "[\t]" is for deleting the space characters as well as the tabs from the input. The line that begins with the period catches what is left from the other rules – in this case, what cannot be matched by a rule is considered an error. The line with the "[0-9]+" regular expression is for reading the integers from the user.

The code for the parser specification file (myCalc.y), used by bison, is the following:

```

%{
#include <stdio.h>
#include <stdlib.h>
int yylex();
int yywrap();
void yyerror(const char* msg);
%}

```

```

%token quitCommand
%token NUMBER

```

■ The Make utility and Makefiles

Make is a powerful build automation tool. Makefiles are the configuration files of GNU make, which is the standard implementation of make for Linux, and they help you organise and group various commands and execute them with a single make command. The Makefile we made to support this project involves the execution of one of the following commands:

```
$ make clean
$ make myCalc
$ make lex.yy.c
$ make myCalc.tab.c
```

The “make clean” rule is pretty simple because it just deletes everything the other steps produced – as you may have guessed, neither myCalc.l nor myCalc.y will be deleted. The “make myCalc” generates the executable file whereas the other two commands generate myCalc.tab.c, myCalc.tab.h and lex.yy.c.

```
%left '+' '-'
%left '*' '/'
```

```
%%
```

```
input      : input statement '\n'
            |
            ;
```

```
statement  : expression {printf("The output is
%d\n", $1);}
            | quitCommand {exit(0);}
            ;
```

```
expression : NUMBER    {$$ = $1;}
            | expression '+' expression  {$$ = $1 + $3;}
            | expression '-' expression  {$$ = $1 - $3;}
            | expression '*' expression  {$$ = $1 * $3;}
            | expression '/' expression  {$$ = $1 / $3;}
            ;
```

```
%%
```

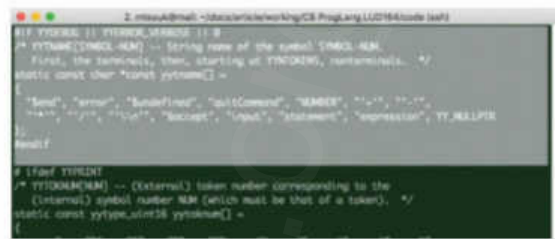
```
void yyerror(const char* msg)
{
    fprintf(stderr, "%s\n", msg);
}
```

```
int main(int argc, char **argv)
{
    yyparse();
    return 0;
}
```

The following rule shows how the addition of two integers is implemented, which is pretty straightforward:

```
expression '+' expression    {$$ = $1 + $3;}
```

The reason for adding \$1 and \$3 is because \$2 is occupied by the ‘+’ operator.



The %left keyword is used for telling that the defined operators are left associative. The definition of the yyerror() function is mandatory but pretty standard.

You can manually process myCalc.l as follows:

```
$ flex myCalc.l
```

The filename of the file generated by flex would be lex.yy.c, which is a finite state machine that recognises the tokens of the language. See the example lex.yy.c in your resources pack.

Similarly, you should process myCalc.y as follows:

```
$ bison -d myCalc.y
```

The filename of the generated file would be myCalc.tab.c, which is a table-driven scanner. The image above shows a small part of the myCalc.tab.c file. If you put the -d command line option when executing bison, bison will also generate a header file that in this case is named myCalc.tab.h. You can use myCalc.tab.h in the flex configuration file to let flex know about the various types of tokens it should expect to process, which is very convenient. What is interesting in myCalc.tab.h is the way the two tokens (quitCommand and NUMBER) are defined:

```
enum yytokentype
{
    quitCommand = 258,
    NUMBER = 259
};
```

This is one of those rare cases where you can see a C enum in action. The image across the page shows the directory with the project files before and after generating the executable file.

Testing the calculator

It is now time to try the calculator. A valid operation with the calculator will produce the next kind of output:

```
32+2+3
The output is 37
12+12*3
The output is 48
```

An invalid operation will produce the following kind of output and exit the calculator:

```
23+123-a
Invalid character!
syntax error
```

So, your calculator is working as expected. ■

BUILD A BETTER WEB

www.webdesignermag.co.uk



web designer

Available
from all good
newsagents and
supermarkets

ON SALE NOW

15 Expert Ruby on Rails tips | Get ready for Angular 2 | Top CSS animation techniques



BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



facebook.com/webdesignermag



twitter.com/webdesignermag

PRACTICAL Raspberry Pi

74 “Scroll through up to five images, which each represent one of the hands that you can play: Rock, Paper, Scissor, Lizard or Spock”



Contents



70 Underwater Pi drone



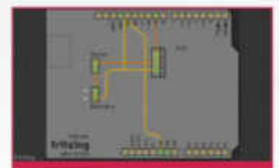
72 Handle OpenGL on your Pi



74 Master Sense HAT skills



78 Turn your Pi into a Tor router



82 Use Fritzing like a pro



Video For the time being, Niels has fitted a GoPro Hero to the front of the ROV for recording purposes. A live video stream is fed to the laptop

Joystick The ROV is controlled via a joystick connected to a laptop. The laptop runs a series of Python scripts, using PyGame to read the signals

3D-printed A few parts have been 3D-printed, as it proved to be the easiest way to tailor the propellers to the exact size and shape needed

Pi The GPIOs are used to help control the four main motors that power the ROV. The Raspberry Pi also relays information back to the server

Components list

- Raspberry Pi B+
- Tilt, roll, pressure and temperature sensors
- 3D-printed camera tilt and propellers
- Camera module
- USB joystick control
- Ethernet cable



Left Pressure sensor data is relayed from the ROV to the laptop as the vehicle dives deeper

Below The camera module is sealed behind a waterproof lens on the front of the ROV





Underwater Pi drone

Niels Affourtit has turned his passion for RC vehicles into a fully functioning, submersible Ras Pi vehicle

Why did you decide to build an ROV (remotely operated vehicle)?

When I was a kid I frequently built RC models; sailing boats and speedboats. While I owned a fantastic RC glider and dreamed of having my own RC submarine someday, my neighbour at the time had an RC submarine and discouraged me from building one by saying this thing always had trouble with its electronics and leakage. Besides, almost all waters are murky in our area!

Around the end of 2014, I saw a *National Geographic* episode about the salvage of the Costa Concordia and was very impressed with how they used a VideoRay to monitor the work done by their divers, and it helped revive my dream of having my own model submarine. That's when I stumbled across the people who have built their own ROV.

How are the inner components of the ROV kept safe from water damage?

Since the Nineties, a lot has changed in the world of RC models. There are now LiPo batteries that give much more energy and power, and brushless outrunner motors that can run submerged in water, making moving parts through the hull unnecessary – this is the key development that makes the chance on leakage very small. The end closure is tightened by external water pressure and happens to be extremely reliable. A lot of the information and lessons I had learned are found on openrov.com, svseeker.com, raspberrypi.org and many other sites with forums about Linux and Raspberry Pi.

How long has the development of the ROV taken? Were there any problems?

When I started in December, I had a lot of stuff to learn and decided that purchasing a low-cost driver was where I should start. I opted for a Raspberry Pi instead of the BeagleBoard used by OpenRov, or an Arduino board. I had previously learned a little Linux from the workstation my

company bought to do stress calculations (using our finite element program ANSYS). We also use Linux to improve daily interaction with third-party software, doing batch runs and fatigue calculations on steel pipelines. For all the functions on this ROV, there are so many options that I always face new challenges; this is what makes this project so interesting.

What role does the Raspberry Pi take in this project?

The ROV is controlled with a joystick connected to a laptop. The laptop runs Python scripts, and with PyGame I can read the signals from the joystick. The signals are then translated into servo commands and sent to the Raspberry Pi via a simple socket connection. The Raspberry Pi is the brain of the ROV; it communicates with the surface laptop via Ethernet. Thanks to the OpenROV project, I learnt to implement a Tenda home plug, which reduces the communication lines from four to two wires, increases the reach from about 50 metres with a submerged CAT5 to 300 metres, and makes the signal much less susceptible to noise.

A battery-powered Wi-Fi router (bought for €1 at the local recycling store) then sends the signal to my laptop. The advantage of the Wi-Fi router is that I can attach it to a reel, avoiding the use of sliding contacts. It also reduces tripping risks on shore. The Wi-Fi router can be mounted on a buoy so that the ROV can be launched from a boat. Signals from the ROV are communicated via a web interface, the same way as OpenROV. All sensor data is written to a MySQL database and an Ajax script reads the data out and presents it on a web page. This web page also presents the live video feed from the camera of the ROV. Currently I'm using a Ras Pi cam, but I have ordered the USB camera used by OpenROV to improve the images.

The GPIO (General Purpose Input and Output) of the Pi is used to control the standard ESC (Electronic Speed

Controller) of the four motors by PWM (Pulse Width Modulation), the tilt servo of the camera, relays for the lights, and server communication. Although the Pi now has many libraries for popular sensors, I still need an Arduino to communicate with the depth sensor (MS5803-5). This Arduino is connected via the USB of the Pi and its analogue port is used to measure the voltage of the battery (using two resistors in series to drop the voltage to a safe 2-3V).

Since there is little space inside the ROV, the standard (USB) power connection on the side cannot be used. Therefore I inject power via the GPIO board at the 5V pin. This isn't a fused connection, imposing some risk, but it has worked perfectly for me.

Why did you decide to use 3D-printed parts in your ROV?

The propellers, tilt mechanism of the camera, and the Raspberry Pi housing are 3D printed. My neighbour imports Inno3D printers and I borrowed one to use for my project. This way I could vary the size, pitch and direction of the propellers. The parts take half an hour to print but cost only a few cents. Later on I will use the printer to make hydrodynamic motor mounts.

How deep can the ROV currently travel?

The tested depth is 13 metres. I took it down to a lake without the electronics and lowered it packed with lead. The collapse pressure determines the maximum depth of the ROV. I design offshore pipelines at INTECSEA and collapse pressure calculations are my everyday work. I made a simple finite element model of the ROV and applied external pressure to see how it would behave. Currently the 6mm-thick end caps are the bottleneck. At 30-40 metres they deflect so much that I anticipate they can start tearing. The lakes in the Netherlands are normally a few metres deep, at a few places 15-25 metres. Even the North Sea here is only 20-30 metres deep, so extending the depth limit has not been my priority. ■



Niels Affourtit

is an avid scuba diver who has turned his love for all things nautical into constructing an underwater ROV.

Like it?

Niels has uploaded a series of YouTube videos depicting the original testing of the ROV, as well as how it performed in open water conditions. Check out his channel: youtube.com/user/gravitylost1

Further reading

Niels took a lot of inspiration from the OpenROV team, who have been producing underwater exploration vehicles for a number of years. Take a closer look at what they do over at www.openrov.com

Handle OpenGL on your Pi

The Raspberry Pi includes a basic GPU. Learn how to use it to generate 3D graphics on board



Joey Bernard

is a true renaissance man, splitting 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

Many of the examples we have looked at in this column have been programs that run on the console. This month, we will be taking a look at how you can include 3D graphics within your Python code. The hardware on the board includes a GPU, so you should be able to take advantage of everything this computer is capable of. Since we are going to be playing with graphics, this means that you will probably want to have an actual monitor connected to your Raspberry Pi. If you don't have one, you can connect to your Raspberry Pi over SSH with X11 forwarding turned on. As long as you have an X11 server on your desktop, you can see the display on your local machine. To get access to OpenGL from Python, you will need to install the correct package, with:

```
sudo apt-get install python-opengl
```

This is assuming that you are using a Debian-based distribution, such as

categories of helper functions available. These are grouped under the GLU and GLUT sub-modules. The imports that will likely be part of your boilerplate are:

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
```

All of the functions that you will likely ever need should be imported to your namespace. One of the really useful functions that GLUT provides is the ability to create windows and to draw within them. In order to use it, you need to initialise the subsystem first:

```
glutInit()
glutInitDisplayMode(GLUT_RGBA |
GLUT_DOUBLE | GLUT_ALPHA | GLUT_
DEPTH)
```

The first thing you will need is a window to draw within. Windows need a size and a position to define where they should

be drawn on the physical screen. A basic window would be defined by:

```
def draw1():
    glClearColor(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    # Other drawing
    glutSwapBuffers()
```

The first line blanks out the contents of the window. The second line resets the position of the window, ready for further drawing. The commented line is where you would place any other drawing commands needed by the draw1 function. The last line is important for double-buffering. The technique of double-buffering is where you draw on an in-memory buffer region, and then dump the entire contents of this region to the physical display device. This is done to get the fastest possible drawing speed. Once you have your function defined, you can set it as the callback function with the GLUT function:

```
glutDisplayFunc(draw1)
```

The last step is to tell OpenGL to start drawing. You can do this with:

```
glutIdleFunc(draw1)
glutMainLoop()
```

The first line tells OpenGL to run the draw1 function whenever the system is otherwise idle. In this case, this means all the time. The second line starts the processing loop for the OpenGL process. You now have a black window, which is rather boring.

Now we need to actually start drawing within this new window. In OpenGL, drawing simple objects is done by using vertices and connecting lines between them. As a simple example, we will look at drawing a basic rectangle. Any drawing commands need to be bracketed by a **begin** command and an **end** command

The actual drawing commands consist of telling OpenGL where the vertices are located within the window

Raspbian. It will also install the OpenGL libraries that actually talk to the GPU and handle the 3D operations. For those who don't already know, OpenGL is the open source, multi-platform library to handle 3D graphics. For those of you coming from the Microsoft world, it is essentially Direct3D for everything non-Microsoft. All of the actual functionality is encoded within libraries written in C, while python-opengl provides a set of Python wrappers allowing you to access these libraries from your code.

The first step in trying to use OpenGL in Python is to import all of the code that you will need. Since OpenGL can be a bit messy and convoluted, there are several

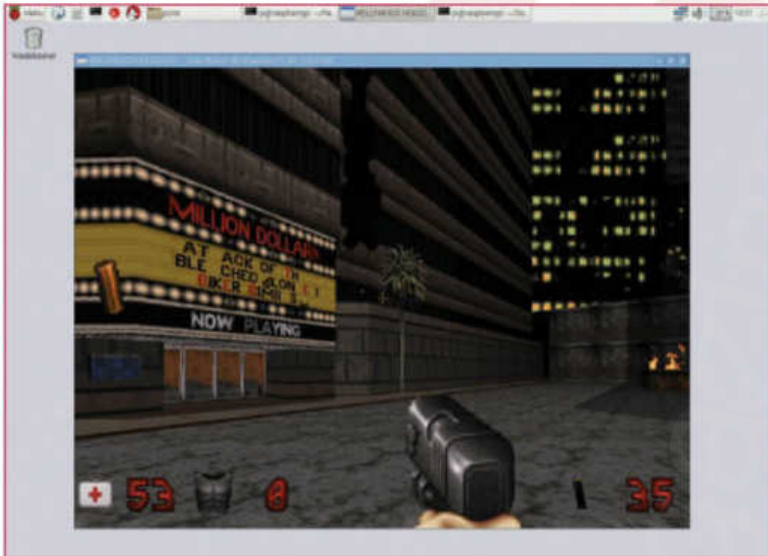
be drawn on the physical screen. A basic window would be defined by:

```
glutInitWindowSize(500, 400)
glutInitWindowPosition(0, 0)
```

The last step is to actually create the window. This creates a new window object and displays it on the display device in use. You can do this with:

```
window1 = glutCreateWindow('window1')
```

The newly created window is created with a title on the main window bar defined by the string handed in as an option. At this point we have a window, but there



Above Raspbian's OpenGL driver means we'll be able to play 3D games more advanced than this!

to tell OpenGL what functions need to be paid attention to. This looks like:

```
glBegin(GL_QUADS)
# drawing commands
glEnd()
```

The actual drawing commands consist of telling OpenGL where the vertices are located within the window. For this simple case, the coordinate system of the window starts at (0, 0) in the bottom left corner. The x-coordinate increases towards the right, and the y-coordinate increase towards the top. If we have the coordinates of the bottom-left corner of the rectangle in the variables x and y, with a width and a height defined, we can set the vertices with the code:

```
glVertex2f(x, y)
glVertex2f(x + width, y)
glVertex2f(x + width, y + height)
glVertex2f(x, y + height)
```

Notice that you need to define the vertices in order, going around your rectangle. You probably also want to draw your rectangle is some other colour than the black background of the window. You can do this

by changing the colour used, just before you start drawing. The command is:

```
glColor3f(0.0, 0.0, 1.0)
```

The parameters are floating point numbers, from 0.0 to 1.0, to define how much red, green and blue to use in the colour. So the above command will give you a bright blue rectangle. If you were to run this code as is, however, you still won't see your rectangle; this is because we haven't told OpenGL that we are drawing in 2D yet. A function that contains all of the required steps looks like:

```
def prepare2d(width, height):
    glViewport(0, 0, width, height)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glOrtho(0.0, width, 0.0, height,
            0.0, 1.0)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
```

Digging into the details of what these functions are and why you need them would probably fill another full article. For now, consider these functions as further reading for the student.

How to handle 3D on OpenGL

"But, what about 3D?" we hear you ask. Luckily, we can reuse much of what we have already covered here to move into three dimensions. The first change is that vertices now are defined with three values rather than two, giving you a set of x, y and z coordinates. The second change is in the category of functions that you need to use. You can see that the drawing functions used are explicitly two-dimensional in their names.

For example, you should have noticed the '2' in names like 'glVertex2f()'. To draw three-dimensionally, this changes to names like 'glVertex3fv()'. The third change is more conceptual. Once we move to three dimensions, we need to worry about where we are in relation to the object that is being drawn, and what the field of view and depth of clipping panes need to be. The essential definitions can be set with:

```
gluPerspective(45, (width/height), 0.1, 50.0)
```

The first parameter is the field of view in degrees, the second parameter is the aspect ratio, and the last two parameters are the near and far clipping panes. OpenGL needs this to figure out how to project your three-dimensional object onto a two-dimensional surface (the screen) so that it looks correct. You start off located right next to the object, which is too close to see it clearly. Luckily, you can move relatively easily. There is a function that can apply a transformation matrix to objects within OpenGL. In order to move back, say five units, you would use:

```
glTranslatef(0.0, 0.0, -5)
```

There are also other types of transformations, like rotations, which can be applied within OpenGL. If we extend the example rectangle to a cuboid, we could draw it the same way by going from vertex to vertex. We can then apply a rotation to it with:

```
glRotatef(1, 3, 1, 1)
```

This takes an angle and x, y and z coordinates and rotates the matrix by that much. You can then clear the screen and redraw the rotated cube. As you can see, we've barely scratched the surface of what you can do with OpenGL in Python – we recommend further study!

Master essential Sense HAT skills

Develop your Sense HAT skills and start a coding a version of Rock, Paper, Scissors, Lizard, Spock



Dan Aldred is a Raspberry Pi Certified Educator and a Lead School teacher for CAS. He recently led a winning team of the Astro Pi secondary school contest

Last November, two Raspberry Pi computers were each fitted with an Astro Pi, which is an add-on board boasting an array of sensors and an 8x8 LED matrix. These were flown to the ISS and delivered to Major Tim Peake with a selection of school children's experiments. In late December, the Astro Pi was released for sale to the public as the rebranded 'Sense HAT'. This consists of exactly the same hardware and sensors set found on the Astro Pi but with a new 'Sense HAT' API. The first part of this two-part tutorial introduces you to the Sense HAT hardware and walks you through the skills required to create a Sense HAT version of the updated classic Rock, Paper, Scissors, Lizard, Spock game. These skills also stand alone and you can adapt them for use in your own projects.

What you'll need

- Raspberry Pi
- Sense HAT

Below The Sense HAT isn't just about the LED array – it's also packed full of useful sensors

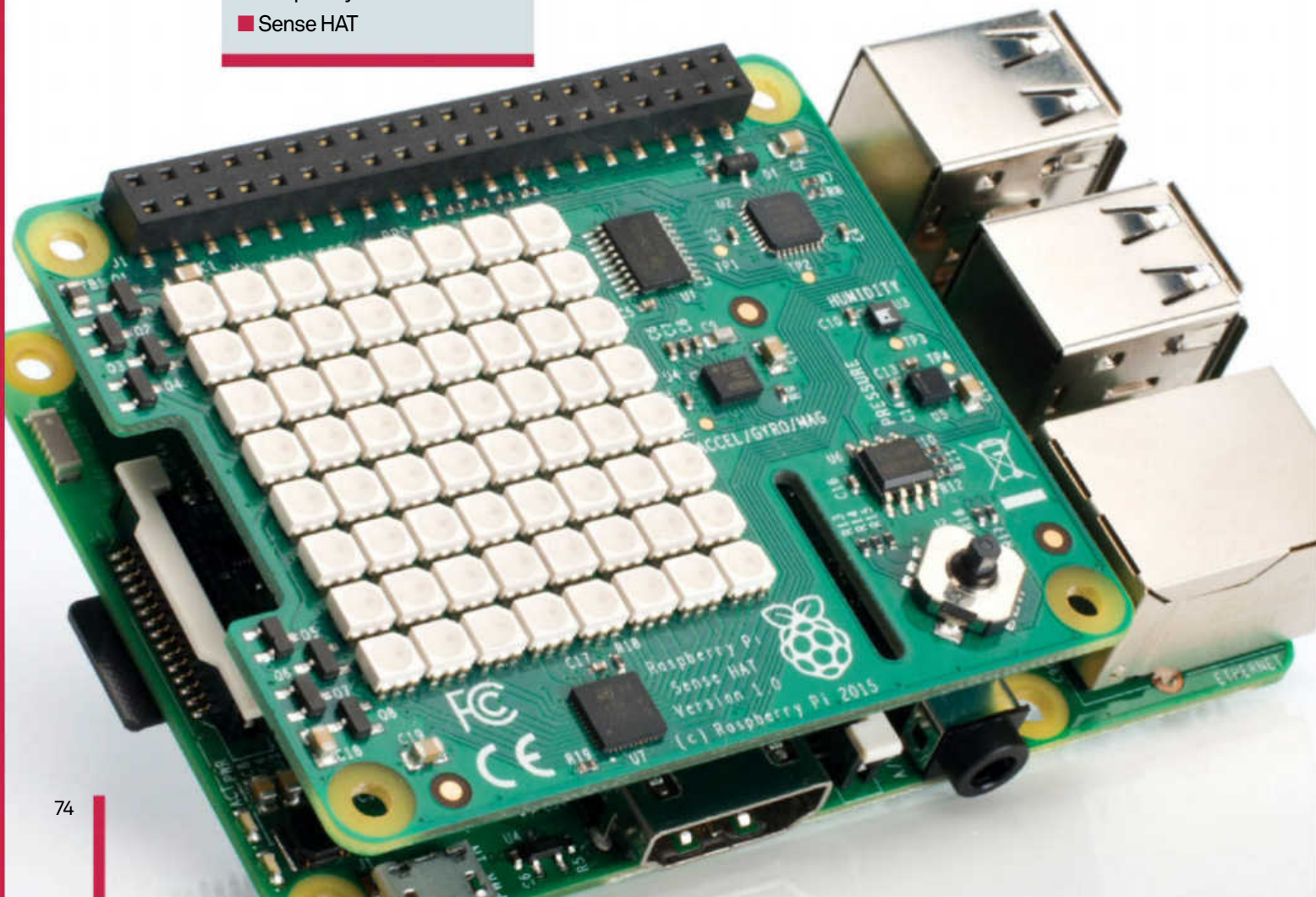
01 Install the Sense HAT software

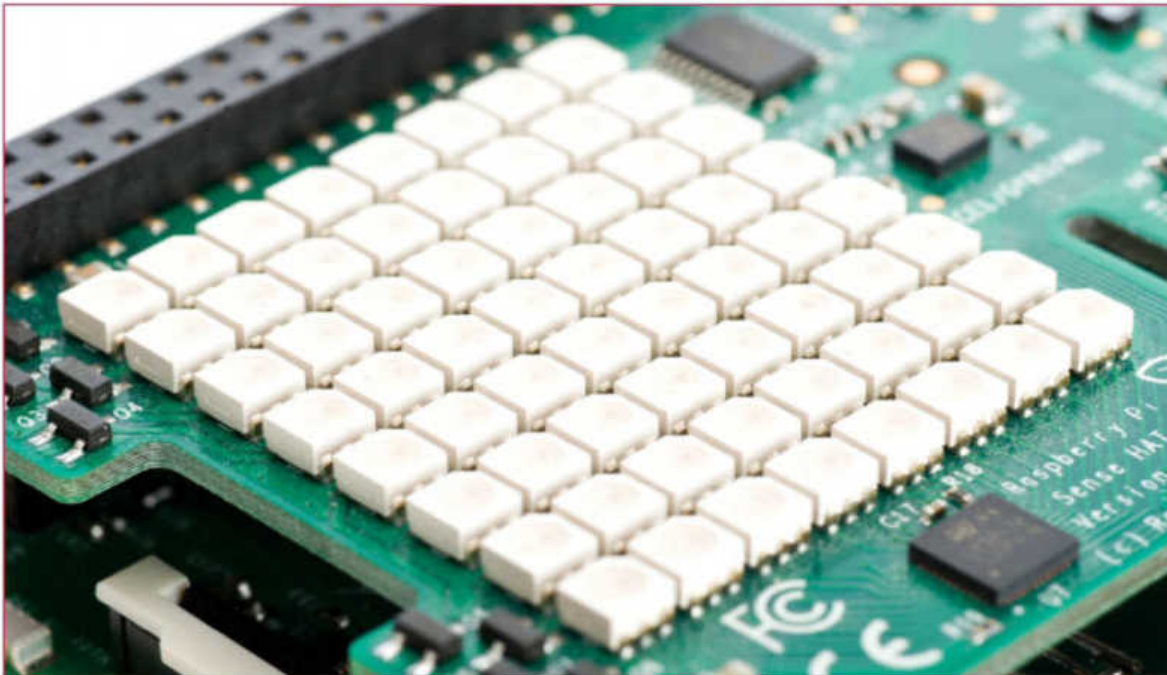
First, attach the board to the GPIO header and install the Sense HAT software. This is pre-installed on the latest Raspbian images; for older images it can be downloaded – boot up your Raspberry Pi, load the LX Terminal and type in **sudo apt-get install sense-hat** to install the software. On completion reboot your Raspberry Pi.

```
sudo apt-get update
sudo apt-get install sense-hat
sudo reboot
```

02 Scrolling a message

Writing code to scroll text on LCD / LED displays can be challenging and frustrating. The Sense HAT API removes the difficulties and simplifies the whole procedure to a simple line of code: **sense.show_message("This is a test message")**. Open your Python editor and enter the code at the bottom of this step,





Left Each of the LED squares in this grid can be individually controlled, so you can draw images

save and then run it. Your message will be scrolled across the Sense HAT LEDs. Change the text between the quotation marks to add your own message. Adjust the colour of the message and the time it takes to scroll by including the lines `text_colour=[255, 0, 0]` (setting the RGB value) and `scroll_speed=(0.05)`. Try experimenting with the example code below:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.show_message("Linux User and Developer", text_
colour=[255, 0, 0])
```

03 Taking a temperature reading

The Sense HAT has a built-in heat sensor that can be used to read and return the current temperature (line 3). The sensor is fairly close to the CPU and this may pick up some of the residual heat. However, on the whole the reading is sound. To measure the temperature, return to your Python editor and type in the code below, then save and run the file. This will return the current temperature reading and print it out:

```
from sense_hat import SenseHat
sense = SenseHat()
temp = sense.get_temperature()
print("Temperature: %s C" % temp)
```

04 Compass reading

One of the more nifty sensors is the magnetometer, which can be used as a compass. This returns a measurement of the Sense HAT's position in relation to magnetic north. Again the code is easy to use: `sense.get_compass()` in line 3 returns the position, which is stored in a variable called `north`. The value that is measured is then printed out in line 4. Use the code example below to test the compass sensor and the readings:

```
from sense_hat import SenseHat
sense = SenseHat()
north = sense.get_compass()
print("North: %s" % north)
```

05 Mapping an LED image from a picture

Images are built up of pixels which combine to create an overall picture. Each LED on the matrix can be automatically set from an image file. For example, an image of a lizard can be loaded, the colours and positions calculated, and then the corresponding LEDs enabled. The image needs to be 8 x 8 pixels in size so that it fits the LED Matrix. Download the supplied test picture file, `lizard.png`, and save it into the same folder as your program. Use the code below to open and load the image of the lizard (line 3). The Sense HAT will do the rest of the hard work for you:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.load_image("lizard.png")
```

06 Make your own 8 x 8 image

There are two further methods to create an image with the LEDs. The first is a superb on-screen program that enables you to manipulate the LEDs in real time. You can change the colours, rotate them and then export the image as code or as an 8 x 8 PNG file.

Install 'Python PNG library' by opening the terminal and typing `sudo pip3 install pypng`. After this has finished, type `git clone https://github.com/jrobinson-uk/RPi_8x8GridDraw`.

Once the installation has completed, move to the RPi folder with `cd RPi_8x8GridDraw`, then type `python3 sense_grid.py` to run the application.

07 Create and export your image

The Grid Editor enables you to select from a range of colours displayed down the right-hand side of the window. Simply choose the colour and then click the location of the LED on the grid, select 'Play on LEDs' to display the colour on the Sense HAT LED. Clear the LEDs using the 'Clear Grid' button and then start over. Finally, when exporting the image, you can either save as a PNG file and apply the code in the previous step to display the picture, or you can export the layout as code and import that into your program.

The origins of RPSLS

Sheldon Cooper from *The Big Bang Theory* famously brought the game into the mainstream media when he explained how it was played to his friend, though it wasn't actually invented on the show. The original expansion of the classic Rock, Paper, Scissors game is attributed to Sam Kass and Karen Bryla.

Python version of RPSLS

If you can't wait for the next part of this tutorial, then you can play around with this Python version of the RPSLS game. The Sense HAT version uses similar mechanics to calculate the winner and losers of each round played: <https://trinket.io/python/46302ff1af>

08 Code each LED: Part 1

The second method to create an image is to individually code each LED and combine these. To set the colour of an LED, create a variable for a colour and assign an RGB value to it. Add additional colours by creating additional variables. Now create a representation of the image using the variable names: in this example, the X and O symbols combine to make a question mark. Set the LEDs with the code `sense.set_pixels(question_mark)`:

```
from sense_hat import SenseHat
sense = SenseHat()
X = [255, 0, 0] # Red
O = [255, 255, 255] # White
question_mark = [
    O, O, O, X, X, O, O, O,
    O, O, X, O, O, X, O, O,
    O, O, O, O, O, X, O, O,
    O, O, O, O, X, O, O, O,
    O, O, O, X, O, O, O, O,
    O, O, O, X, O, O, O, O,
    O, O, O, O, O, O, O, O,
    O, O, O, X, O, O, O, O
]
sense.set_pixels(question_mark)
```

09 Code each LED: Part 2

To add more colour variations to your image, create a new variable and assign the RGB values using the same format. Replace your previous code with the example below to create a new image. Lots of websites can convert the RGB values for all 16,777,216 colours – try <http://www.colorpicker.com>.

```
X = [255, 0, 0] # Red
O = [255, 255, 255] # White
B = [0, 255, 0] #Green
new_image = [
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X,
    B, B, O, O, O, O, X, X
]
sense.set_pixels(new_image)
```

10 Create an image of a lizard

Now you can begin to customise your own images to use. To get started, create a simple lizard using the LED layout. Then experiment with your own ideas and concepts. Remember, you will also need images for Rock, Paper, Scissors and Spock. Johan Vinet has some excellent and inspirational examples of 8 x 8 pixel art <http://johanvinet.tumblr.com/image/127476776680>.

11 Selecting a random image

Next you need a code to scroll through your images. Begin by renaming each of the images starting from one – for example 1.png, 2.png. Create a new variable called `playersChoice` which will store the current picture filename. Next, load the image using the code `sense.load_image(str(playersChoice) + ".png")` – note that `playersChoice` needs to be converted into a string before it is used. Once the picture is displayed, the variable is incremented, which loads the next picture. Finally, add a conditional to check if you have reached your last picture. For example, if you have five images then use:

```
if playersChoice == 5:
    playersChoice = 0
```

... to check when the variable reaches the last image and to reset the variable to zero. This loads the first picture and the loop starts again from the beginning:

```
import time
while True:
    sense.load_image(str(playersChoice) + ".png")
    playersChoice = playersChoice + 1
    time.sleep(1)
    if playersChoice == 5:
        playersChoice = 0
```

12 Clearing the image

When cycling through the images, the LEDs will turn on and off to the corresponding requirements of your code. However, there will be times when you need to turn off all the LEDs at once. This is referred to as clearing them. To clear the LEDs and set them all to 'off', use the line `sense.clear()`.

13 Setting the joystick up in PyGame

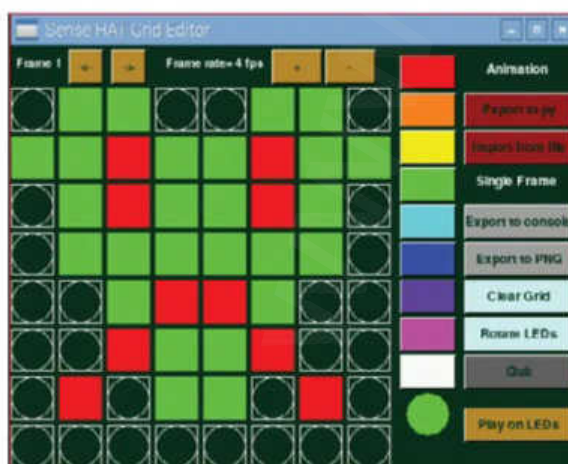
The Sense HAT is equipped with a small multi-directional joystick that can be programmed to respond to direction. The RPSLS game makes use of the joystick to cycle through your images and to select one. In a new Python file, set up the PyGame window by importing the PyGame modules at the top of your file (lines 1 and 2). Next, initialise with `pygame.init()`. PyGame runs in a separate window, but since the game takes place on the Sense HAT, the window is obsolete. You can therefore minimise the size by using: `pygame.display.set_mode((140, 180))` line 5.

```
import pygame
from pygame.locals import *

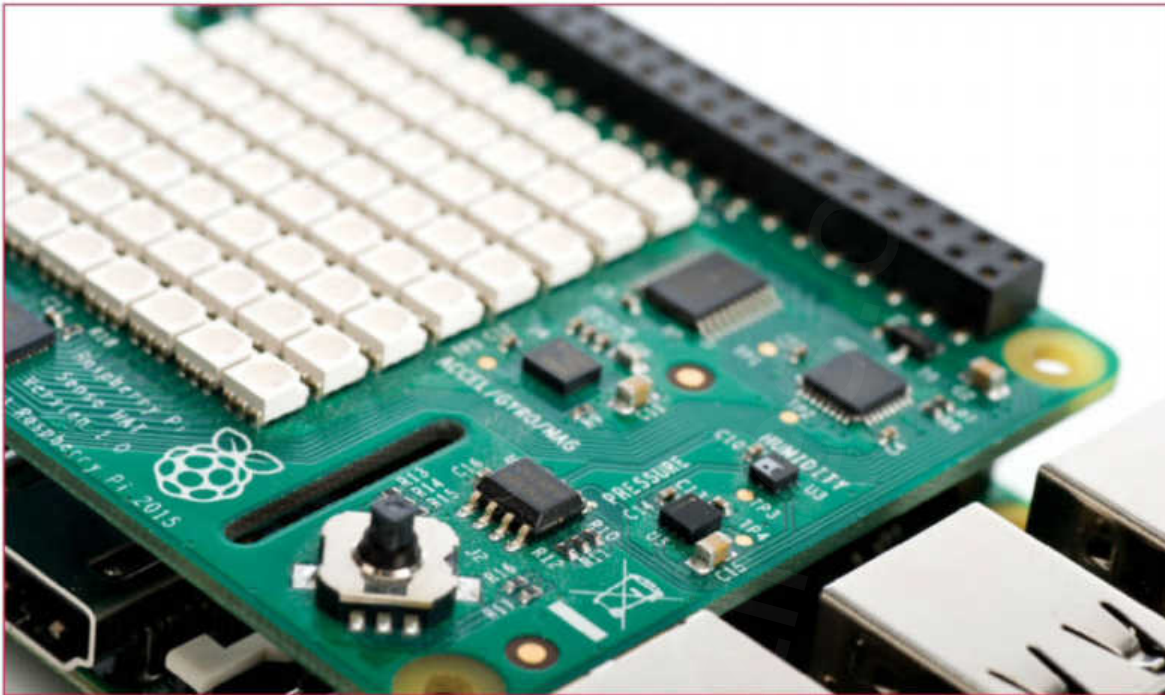
###Set up PyGame Screen###
pygame.init()
pygame.display.set_mode((140, 180))
```

14 Add joystick controls

To add the joystick movement, create a variable called "running" which is set to True (line 1). Then add a `while` statement (line 2) to continually check whether the joystick has



Right The Sense HAT Grid Editor makes it easy to generate your image code



Left We'll use the tiny joystick in the bottom-right corner as the input for our selection interface

been moved. The program checks for an event on line 4 and the picture cycle restarts after the fifth image. Line 5 checks for two conditions being met: a key being pressed and the `playersChoice` being less than a value of 5. If the condition is met, it checks for a joystick movement. The up arrow on the keyboard corresponds to `K_UP`, which in essence is the joystick moved up (line 6). Finally, add a print statement to test that code is responding to the joystick movement and that it works correctly (line 7).

```
running = True
while running == True:

    for event in pygame.event.get():

        if event.type == KEYDOWN and playersChoice < 5:
            if event.key == K_UP:
                print("UP")
```

15 Selecting a picture

You have now coded PyGame to wait for 'events' and check and respond if the joystick is moved 'up'. In the RPSLS game, this procedure is used to enable the player to select a picture that represents their turn – for example, Paper. Replace the `print("UP")` on line 7 with the image section code used in step 11. As before, use `sense.load_image(str(playersChoice) + ".png")` to load the image onto the LEDs, then increment the `playersChoice` variable using `playersChoice = playersChoice + 1` to select the next picture. Line 6 checks if you have reached the end of the picture cycle and then resets it back to a value of zero (line 8), so the cycle begins again starting from picture one:

```
if event.type == KEYDOWN and playersChoice < 5:
    if event.key == K_UP:
        print (playersChoice)
        sense.load_image(str(playersChoice) + ".png")
        playersChoice = playersChoice + 1
    if playersChoice == 5:
        playersChoice = 0
```

The Sense HAT is equipped with a small multi-directional joystick that can be programmed to respond to direction

16 Confirming your selection

Once you have scrolled through and chosen your picture then you need to be able to select it. This is achieved pressing the whole joystick button down, which acts as if the Return key has been pressed. On line 2, the code `if event.key == K_RETURN`: checks for this action and then responds by breaking the loop. The loop stops cycling through and the current image number is then stored inside the `playersChoice` variable, where it is used later in the game to compare with the Computer's choice and calculate the winner.

```
```Checks for a 'select / Enter' Choice```
 if event.type == KEYDOWN:
 if event.key == K_RETURN:
 running = False
 break
    ```Ends loop and moves onto main game```
```

17 RPSLS

You now have a basic structure for the start of the RPSLS game. The program enables you to scroll through up to five images, which each represent one of the hands that you can play: Rock, Paper, Scissor, Lizard or Spock. In the next issue, part two of this tutorial will code the mechanics of the gameplay and combine this to create the finished RPSLS game. A sneak video of the final game in action can be viewed here: www.youtube.com/watch?v=T_ZvWkMgVFM. In the meantime, work on creating your 8 x 8 pixel art. ■

Anonymise your web traffic with a Pi Tor router

Use your Raspberry Pi as a wireless access point and route traffic anonymously



Christian Cawley

is a former IT and software support engineer. Since 2010 he has written for computer and smartphone users, both online and in print magazines

Are you concerned about your online privacy? Suspect that the NSA or GCHQ are tracking you online? Or are you simply anxious about advertising networks targeting their ads at you? If this is the case, or you're simply on the lookout for a new project, then setting your Raspberry Pi up as an anonymising Tor wireless access point is a good way to save time and effort setting up Tor on all of your PCs and mobile devices.

If you're unaware, Tor is free software that enables anonymity online and routes internet traffic through a worldwide anonymising network of over 7000 relays. By doing this, your location and website use can be concealed from techniques such as traffic analysis and network surveillance.

This is all about protecting your privacy by making your visits to websites, instant messages, emails and more difficult to trace, and can be set up on your Raspberry Pi with a wireless dongle (or a Raspberry Pi 3), Ethernet cable, and a bit of configuration.

01 Connect everything

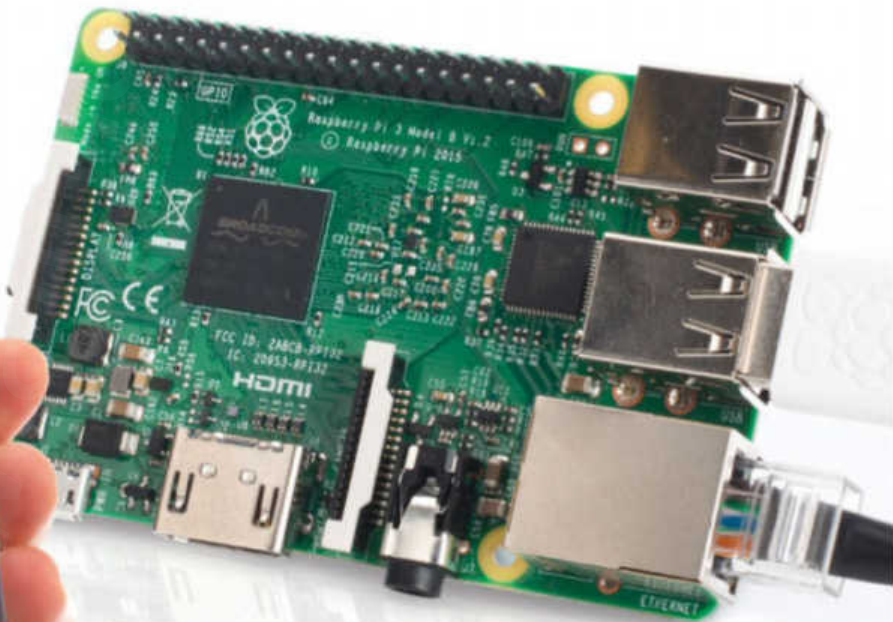
With everything connected and switched on, you can begin configuring the Pi router. So begin by plugging one end of the Ethernet cable into your Raspberry Pi and the other into a spare port in your router.

If you're using a pre-2016 model of the Raspberry Pi, connect a wireless dongle to the computer. Raspberry Pi 3 owners don't need to do this as the computer has built-in wireless networking.

02 Remotely configure with SSH

The next step is to connect to your Raspberry Pi via your PC, and you can do this simply using SSH. (If this isn't an option, however, continue the rest of the tutorial with a monitor connected to your Raspberry Pi, with a mouse and keyboard.) To use SSH, open a terminal and enter:

```
ssh pi@192.168.0.27
```



What you'll need

- Ethernet cable
- Wireless router
- USB WiFi adaptor or Raspberry Pi 3

Left Once you've set your Tor router up, you can browse anonymously from your smartphone if you connect to that router's Wi-Fi signal

```
pi@raspberrypi:~$ ssh pi@192.168.0.27
Warning: Permanently added '192.168.0.27' (ECDSA) to the list of known hosts.
pi@192.168.0.27:~$
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 7 16:17:07 2015
pi@raspberrypi:~$
```

... where the Raspberry Pi has the default username of “pi” and the IP address 192.168.0.27. If your distro doesn’t have SSH installed, you can install openssh-client or putty. You can also communicate via SSH in Windows using PuTTY.

You’ll find the IP address of your Raspberry Pi by checking your router’s admin page, or connecting a monitor and keyboard and entering ifconfig.

03 Install access point

Next, connect to the Raspberry Pi with SSH and enter **iwconfig** to ensure the wireless adaptor is recognised. Then, refresh the package list with **sudo apt-get update** and install the wireless access point software:

```
sudo apt-get install hostapd isc-dhcp-server
```

```
pi@raspberrypi:~$ sudo nano /etc/dhcp/dhcpd.conf
GNU nano 2.2.6 File: /etc/dhcp/dhcpd.conf Modified

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local5;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
```

04 Edit DHCP settings

Next, we need to edit the /etc/dhcp/dhcpd.conf file, to avoid IP addresses being assigned randomly. Open in nano with:

```
sudo nano /etc/dhcp/dhcpd.conf
```

Scroll through for the lines starting “option domain-name” and comment them out with a # as follows:

```
#option domain-name-servers ns1.example.org, ns2.
example.org;
#option domain-name "internal.example.org";
```

Following this, find “authoritative” and uncomment the line by removing the #.

05 Specify IP addresses

Head down to the end of the document and add the following lines to define the IP address for the access point, the range of addresses and the domain name servers (also known as DNS):

Tor is free software that enables anonymity online and routes internet traffic through a worldwide anonymising network

```
pi@raspberrypi:~$ nano /etc/dhcp/dhcpd.conf
GNU nano 2.2.6 File: /etc/dhcp/dhcpd.conf Modified

# range 10.17.224.10 10.17.224.250;
# }
# pool {
#   deny members of "foo";
#   range 10.0.29.10 10.0.29.230;
# }
# }
subnet 192.168.12.0 netmask 255.255.255.0 {
  range 192.168.12.5 192.168.12.50;
  option broadcast-address 192.168.12.255;
  option routers 192.168.12.1;
  default-lease-time 600;
  max-lease-time 7200;
  option domain-name "local";
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

```
subnet 192.168.12.0 netmask 255.255.255.0 {
  range 192.168.12.5 192.168.12.50;
  option broadcast-address 192.168.12.255;
  option routers 192.168.12.1;
  default-lease-time 600;
  max-lease-time 7200;
  option domain-name "local";
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Note that the domain name servers specified here are provided by Google. Press Ctrl+X to save and exit, following the prompts.

```
pi@raspberrypi:~$ nano /etc/dhcp/dhcpd.conf
GNU nano 2.2.6 File: /etc/dhcp/dhcpd.conf Modified

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local5;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
```

06 Assign a static IP

Because the Raspberry Pi’s wireless card receives a dynamic IP address by default from your router, we need to specify a static IP so that it can always be accessed in its role as a Tor anonymiser.

With nano, open /etc/default/isc-dhcp-server, scroll down to the line that currently reads INTERFACES="", and edit so it reads INTERFACES="wlan0". Save and exit.

Deal with errors

Various errors can occur during the setting up of this project. Problems with your Locale settings are one example, and this can be dealt with by editing the ssh_config file. Enter:

```
sudo nano /etc/ssh/ssh_config
```

Find the line that reads “SendEnv LANG LC_*”, commenting it out with a # at the beginning. You may need to reboot your Raspberry Pi to fully overcome the error.

Which Pi can you use?

While the older, dual USB port Raspberry Pis can be used for this project, the best results are achieved with the Raspberry Pi 2 and 3 models. The reasons are simple: both have four USB ports, thereby enabling the connection of other devices such as keyboard and mouse if necessary, and both are fast enough to deal with the volumes of data routing required. A Pi Zero probably wouldn't be practical as a long-term solution.

07 Set up the wireless adaptor

With wireless adaptor wlan0 now specified, we need to disable it before configuring, using the command `sudo ifdown wlan0`. Using nano once again, open the `/etc/network/interfaces` text file and comment out three lines with #, so they read:

```
# iface wlan0 inet manual
# wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
# iface default inet dhcp
```

Find the line `allow-hotplug wlan0` and add the following:

```
iface wlan0 inet static
address 192.168.12.1
netmask 255.255.255.0
```

Use Ctrl+X to save, and activate:

```
sudo ifconfig wlan0 192.168.12.1
```

08 Configure your new WAP

With the wireless access point physically built, the software installed and then defined, it is time to configure it. In nano, create a new file:

```
sudo nano /etc/hostapd/hostapd.conf
```

To this, add the following:

```
interface=wlan0
ssid=TorHotSpot
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=$Your_Passphrase$
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

You can specify your own SSID; you'll also need to specify your own secure password.

09 Find the configuration file

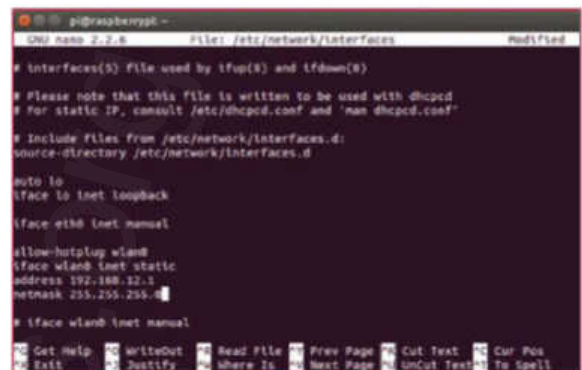
By default, your Raspberry Pi will not be able to see the new configuration file. In order for it to be detected when the wireless access point boots, we need to point to it in the `/etc/default/hostapd` file. To do this, open the file in nano and look for the line that reads `#DAEMON_CONF=""`. Uncomment this, and add the file path to the file:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Press Ctrl+X to save and exit.

10 Configure network address translation

Network address translation, or NAT, needs to be configured to allow multiple devices – laptops, smartphones, tablets, media streamers, etc. – to connect to the Raspberry Pi's wireless access point and route traffic through that single IP address. Open `/etc/sysctl.conf` with nano and at the bottom of the file add:



```
net.ipv4.ip_forward=1
```

Again, press Ctrl+X to save and exit, and enter this command to activate forwarding:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

11 Specify routing rules

With this done, routing rules need to be added. These will connect the Ethernet port eth0 and the Raspberry Pi wireless port wlan0:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

12 Use the rules upon boot

Routing rules will be deleted when you restart the Pi, so in the terminal enter:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

This ensures the rules will persist. Next, open `/etc/network/interfaces` with nano and add this line to the end to load rules when the device boots:

```
up iptables-restore < /etc/iptables.ipv4.nat
```

Save and exit, then restart the DHCP server:

```
sudo service isc-dhcp-server restart
```

Finally, enable the access point:

```
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

13 Ready your access point

At this stage, it should be possible to connect to the Raspberry Pi wireless access point. But you'll need to guarantee some settings, starting with the DHCP and hostapd services:

```
sudo service hostapd start
sudo service isc-dhcp-server start
```

Update the initialisation (init) scripts with:

```
sudo update-rc.d hostapd enable
sudo update-rc.d isc-dhcp-server enable
```

NAT needs to be configured to allow devices to connect to the Raspberry Pi's wireless access point

Restart with `sudo shutdown -r now`, and retry the connection from your mobile device.

14 Time to install Tor

As things stand, the Raspberry Pi is a wireless access point. But it is not anonymised yet. Reconnect via SSH and then run `sudo apt-get install tor`, then open `/etc/tor/torrc` in nano and add the following to instruct Tor to anonymise the wireless access point:

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.12.1
DNSPort 53
DNSListenAddress 192.168.12.1
```

15 Add SSH exception

Next, flush the IP tables with:

```
sudo iptables -F
sudo iptables -t nat -F
```

And follow this by adding an exception for SSH connections on port 22:

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp
--dport 22 -j REDIRECT --to-ports 22
```

These two rules enable DNS lookups and direct TCP traffic to Tor's port 9040, respectively.

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p udp
--dport 53 -j REDIRECT --to-ports 53
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp
--syn -j REDIRECT --to-ports 9040
```

Above You could even use the Pi 3's Bluetooth to control your smart home gadgets as well as route your web traffic through Tor

Save to the NAT file with:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

16 Enable the Tor service

At this stage, you can enable the Tor service on your wireless access point with:

```
sudo service tor start
```

Boot scripts can be enabled with:

```
sudo update-rc.d tor enable
```

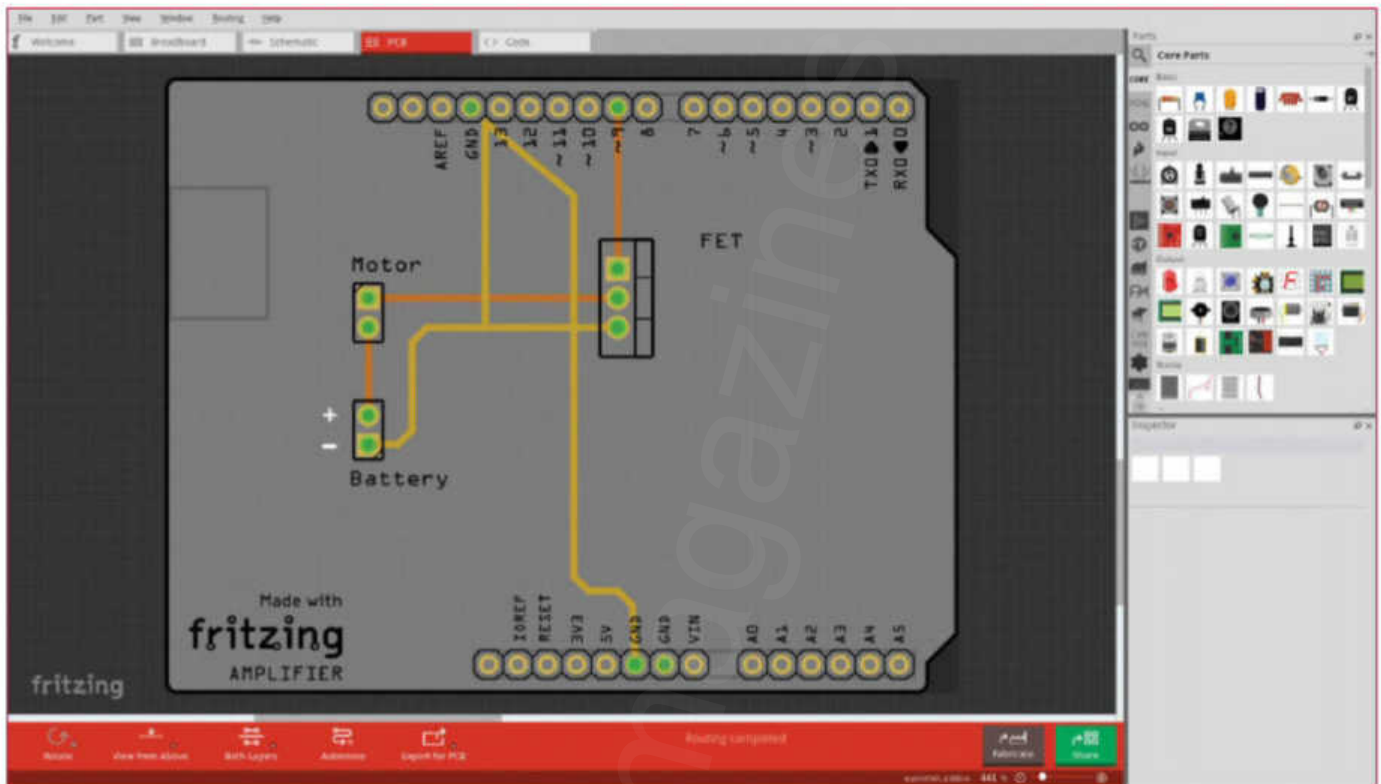
Restart the Pi again with `sudo shutdown -r now`.

17 Verify Tor anonymity

With your Raspberry Pi wireless access point restarted, it's time to connect using a Wi-Fi enabled device, perhaps a smartphone or laptop. As before, the connection should work without any problems – but are you browsing anonymously?

Find out by heading to <https://torproject.org> from two devices and comparing the IP addresses. If they're different, you're anonymous! ■

Create your own circuit diagrams with Fritzing



Christian Cawley is a former IT and software support engineer. Since 2010 he has written for computer and smartphone users, both online and in print magazines

Wondering how we illustrate circuit diagrams? With Fritzing – a donationware utility you can use too!

Putting circuits together can be tricky, even for experts.

Fortunately, various notations exist that make it easier to work out just where wires, components and power sources are connected. But rather than jot these details down with a pencil and paper, you might prefer to bring an element of realism to the illustration. This is particularly useful for beginners, and if you're demonstrating a breadboard-mounted circuit with an Arduino, for instance, you might wish to clearly and accurately illustrate the precise positions of each component and wire. This is where Fritzing comes in.

You've probably seen illustrations of circuits in the pages of **Linux User & Developer** over the years, and with Fritzing you too can put together circuit illustrations, simply by dragging and dropping the categorised components into place, saving, and exporting. In addition to breadboard illustrations, you can also create traditional schematics and PCB diagrams.

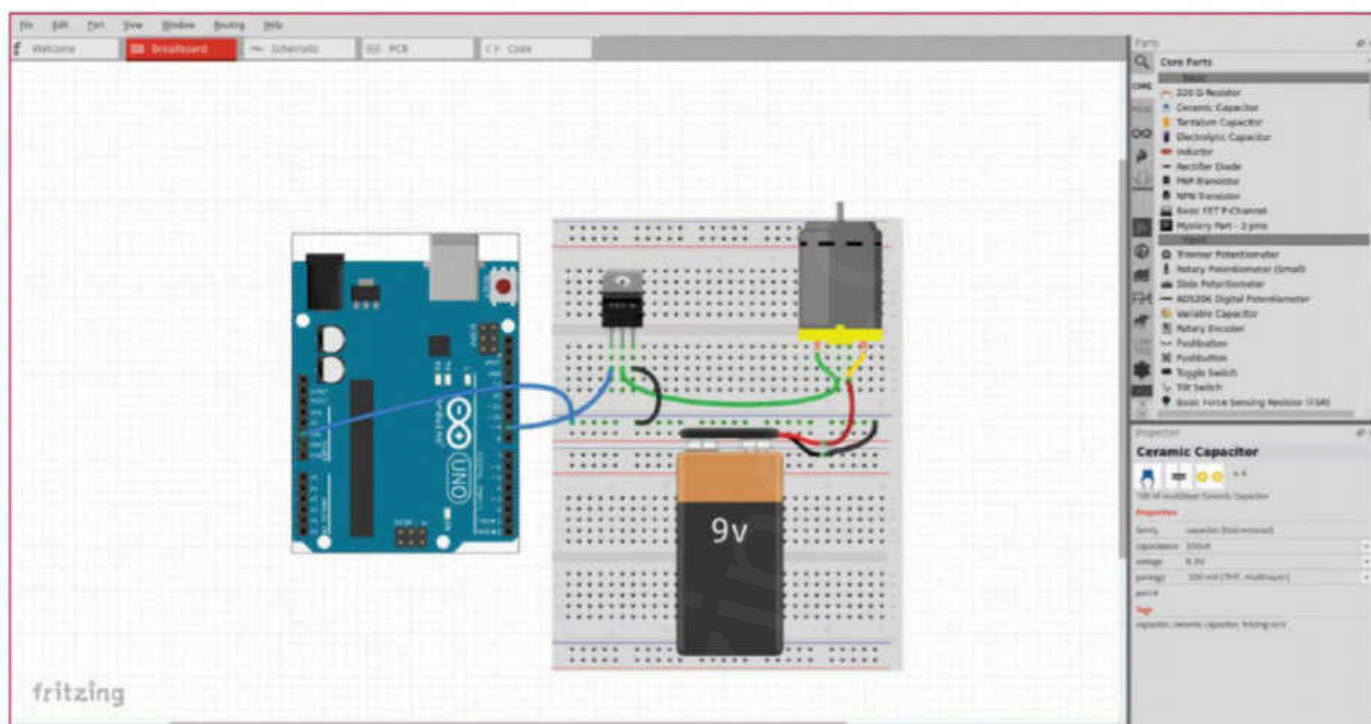
With some of the same familiar shortcuts and tools that you may have already used in an art package, Fritzing is remarkably easy to use.

01 Install Fritzing

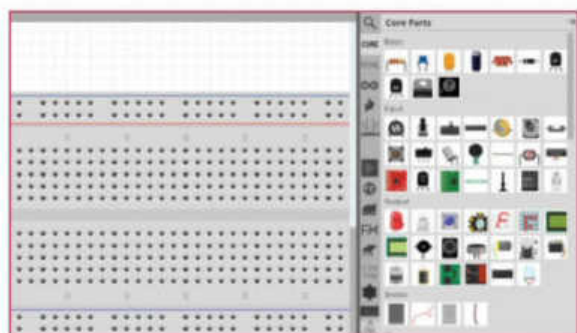
For the quickest installation option, use `sudo apt-get install fritzing` on Ubuntu (or whichever Debian fork you're using). Alternatively, for Fedora use `yum install fritzing`. If you prefer, you can also download from the website (ensuring you select the 32-bit or 64-bit as appropriate for your system) and extract the contents. Once you've done this, navigate to the directory in the Terminal and run `./Fritzing` to launch.

02 Select your project type

With the app launched, you'll need to select an appropriate project type. Three are available in Fritzing: Breadboard, Schematic and PCB. We're starting with the Breadboard project, so select this. Across the bottom of the screen you'll see controls to add notes and rotate the board; on the right, various component illustrations are listed by category. You'll find the IP address of your Raspberry Pi by checking your router's admin page, or connecting a monitor and keyboard and entering `ifconfig`.



Above Fritzing covers a huge range of components as well as boards like the Arduino



03 Select your parts

The next stage is to identify the parts that you need to add to the breadboard to complete your diagram. For a finished diagram you'll want to keep this as simple and clear as possible, but for now you have the scope to have a play around. In the Parts box in the top-right corner, choose the appropriate tab and drag components and wires to your breadboard project.

04 Manipulate your board

Before you begin to add components to the board, spend a few moments with the mouse to get an idea of what you can do. Left-clicking on the board allows you to move it about, for instance, while right-clicking displays a menu with commands to rotate, raise, lower and lock parts, and more. You can also use your mouse roller and the Ctrl key to zoom in and out.

05 Add wires

Adding a wire by left-clicking the start point and dragging to the end point. Breadboard pins have labels assigned, and these can be displayed by hovering your mouse pointer over the pins you wish to identify.

You can also add pivots to the wires, which can be useful if you have a busy project. Simply left-click along the length of the wire and pull it in the direction you want.

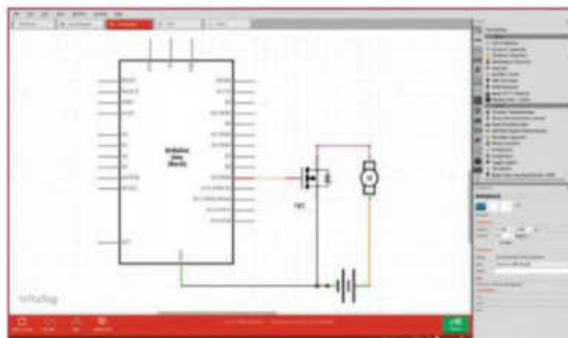
Put together circuit illustrations, simply by dragging and dropping

06 Add components

To build up your project, you'll need to drag the components you need into the main panel. Look carefully as you do this; Fritzing highlights the areas on the breadboard that enable connections to the component. This should aid you with the correct connection of your wires, although it isn't a perfect guide for creating circuits. However, if you're new to breadboards, it can help you understand how they work.

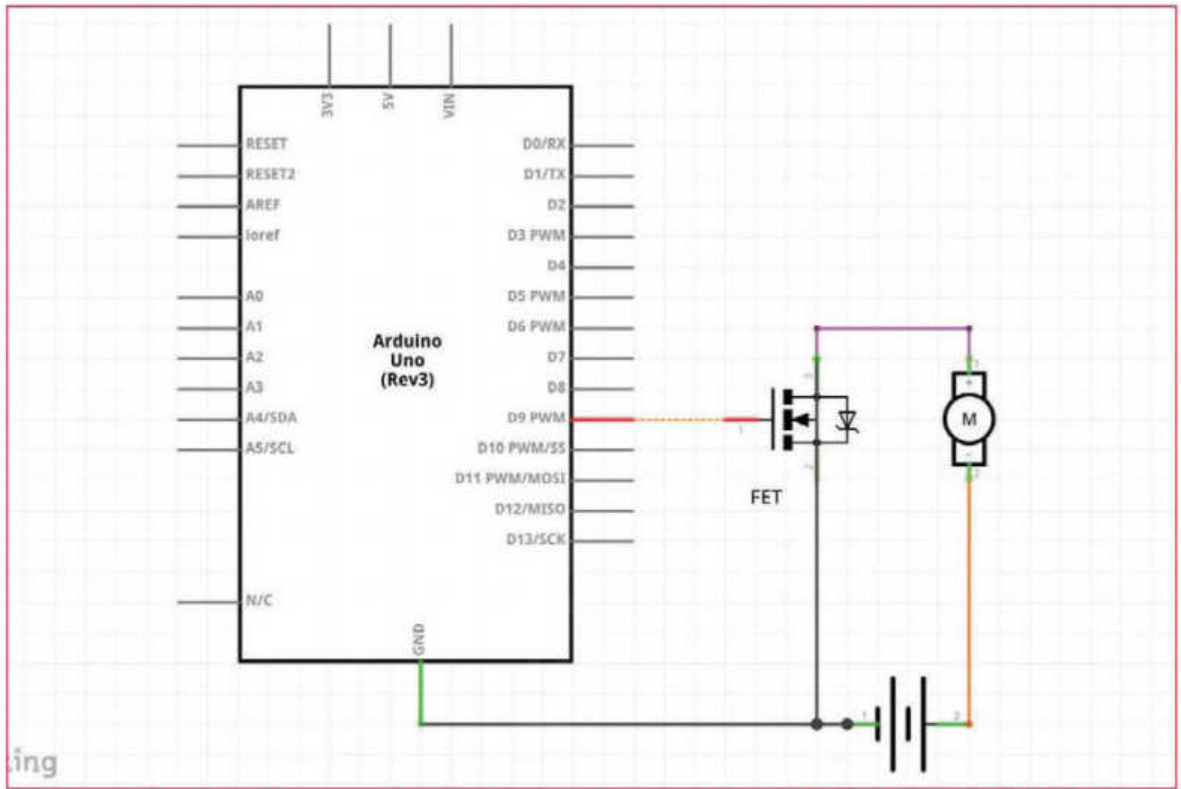
07 Switch views

Breadboarding is an easy way of putting circuits together. But the graphical representation isn't to everyone's liking. Fortunately, Fritzing includes various views that provide alternatives. Across the top of the screen you will notice the Schematic and PCB buttons. Click these to switch views – the project you're currently working on will be displayed in a different format! This can prove useful later on at the export stage.



Get organised

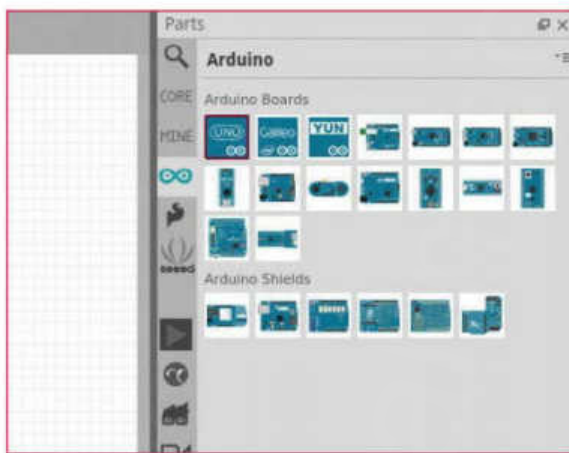
Once you start using Fritzing regularly, it's a good idea to grab all of the components that you intend to use early in the project, and just leave them in the main panel, ready to use. Then, if you need to regularly switch between the different part categories, you can easily pull in the components without wasting time switching between categories – you've already selected them all!



Right You can convert your diagrams into proper schematics, too

Mine your parts

In the Parts panel the second tab is empty when you first run Fritzing. Labelled MINE, this is a place where you can collect the components and boards you use regularly. To add preferred parts to this list, ready for instant access the next time you create a Fritzing diagram, browse to the relevant tab, find the component you want, and drag and drop it onto the MINE tab.



08 Incorporate other boards

So far we've been playing around with

08 So far we've been playing around with the breadboard on its own. But what if you were developing a project with an Arduino and a breadboard-based circuit?

The answer is in the Parts panel on the right. Select the Arduino symbol (a “-” and “+” wrapped in a Möbius strip) and drag the corresponding Arduino board into the main panel, alongside your breadboard. A selection of Arduino shields is also included.

09 Familiarize yourself with Parts

09 In fact, there are so many parts that you can use in a Fritzting diagram that it is quite easy to find yourself bewildered by the choices available to you.

Use the tabs on the left column of the Parts panel to find the items you're looking for. You'll find parts for Arduino, Intel Edison and Galileo, SparkFun and many more.

As you mouseover or select each part, details are displayed in the Inspector panel.

Fritzing helps you to create diagrams; it can also inform you if there is a problem

10 Manipulate other views

10 As you switch through to the alternative views (Schematic and PCB) you'll find that the options for moving things around are limited, although still possible.

For instance, in the illustration above, while the rectangle representing an Arduino can be moved around with a simple left-click and drag, the circuit originally built on a breadboard cannot. Instead, it is merely a collection of components that are connected with wires.

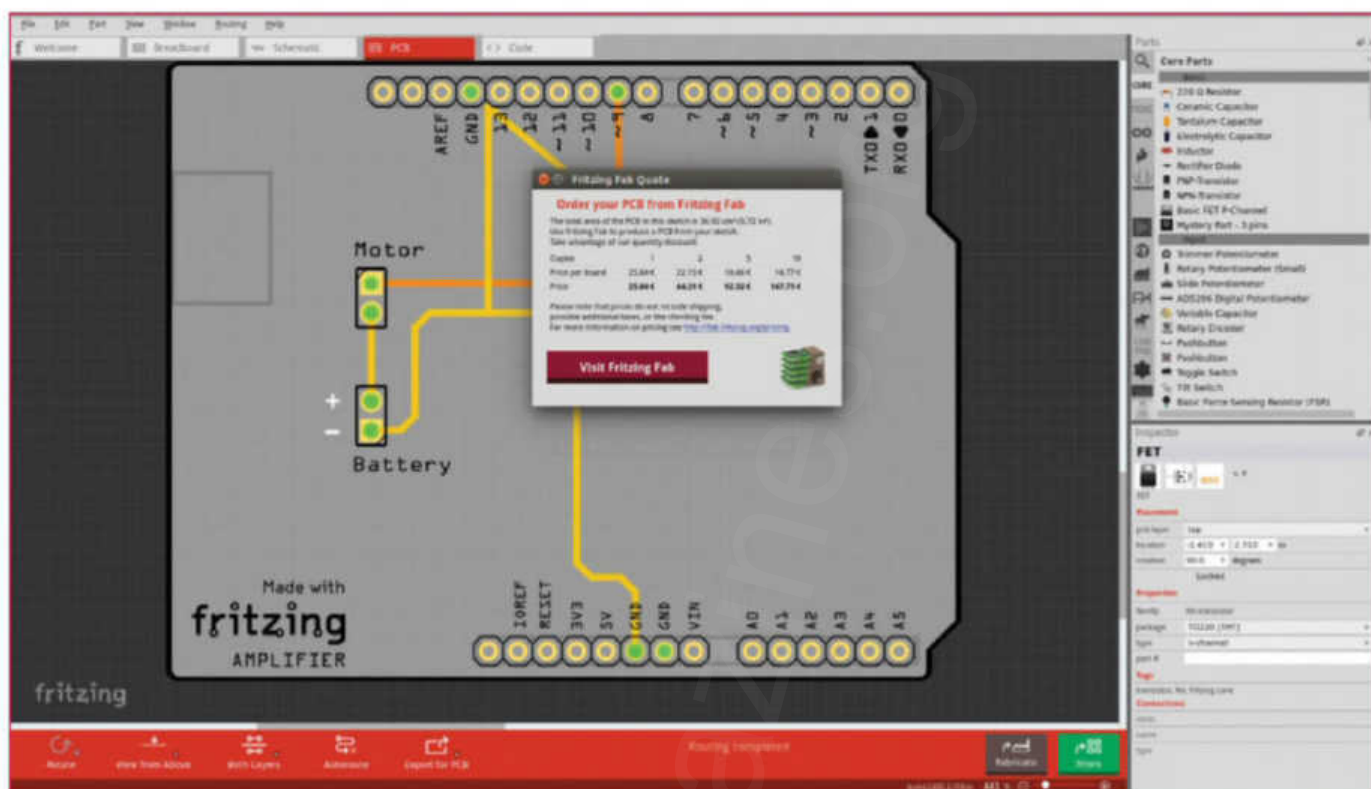
11 Playing with the PCB view

Similarly, the PCB view offers both a different visual experience, and as such how you move things around also changes. Rather than the Arduino and breadboard circuit being presented as separate entities, in this view the circuit sits atop of the Arduino.

Adjusting the position of components is trickier here, but this presents a more intimate look at just how the parts on the breadboard are connected to the Arduino.

12 Use the grid

12 Keeping your project tidy is important, more so your Fritzing diagram. To help here, enable the View > Align to Grid and Show Grid options, which will assist with repositioning your boards. How far the selected items move will depend on which of the three views you are currently using, and your zoom level. Repositioning is via the mouse, and fine-tuning using the arrow keys.



13 Autoroute assistance

While it isn't possible to automate the creation of your Fritzing project, the next best thing is available. While using the Schematic and PCB views, you can activate Routing > Autoroute to automatically connect the components. You can also press Ctrl+Shift+A to get the same results. What this means is that all you need to do is place the components – the autoroute tool will do the rest for you!



14 Is your project ready?

Not only does Fritzing help you to create diagrams, it can also inform you if there is a problem with the project. Once you have your components in place and the wires connected, open Routing > Design Rules Check and let the software assess the state of the project, and report back with problems.

Any issues that come up are usually addressed by adjusting the position of parts and wires.

15 List your parts

Large projects can run away with you. When you're prototyping with Fritzing or on a breadboard, it is easy to overlook something, which is where the Bill of Materials comes in useful.

Found in the File > Export menu, selecting this creates a HTML file that lists the all of the items included in the diagram, which means you have a clear list to refer to for shopping purposes.

16 Get a quote

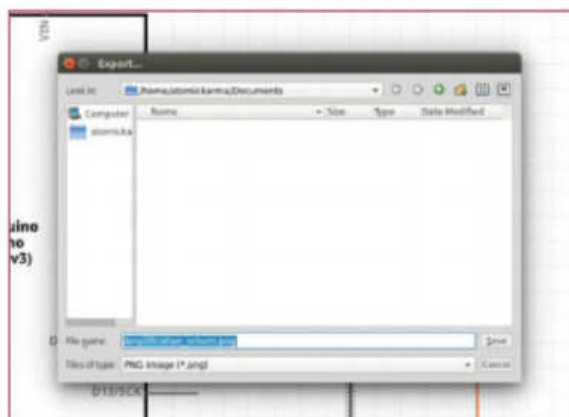
Diagrams can be converted into physical entities using the Fritzing Fab service, which can be initiated via Routing > Fritzing Fab Quote. In the resulting box you will see information about the size of the board, and a quote for 1, 2, 5 or 10 boards to be printed, displayed in Euros. What this means is that a prototype board can be designed in Fritzing and a physical copy easily ordered!

Above If you've designed a PCB – perhaps a new Pi HAT – you can order a prototype from inside the software!

17 Export your Fritzing project

With your Fritzing project complete, you'll probably want to share it with others. Sharing is possible with the Export tool, found in the File menu, where two main options are available: As Image and For Production. If you're planning to print your own circuit, use the latter option.

For sharing a JPG, PNG, SVG or PDF of the diagram, however, select As Image and the preferred format. It's really worth checking out some existing projects, as there's so much you can do with Fritzing that you may not have even thought of. ■



Detailed placement

Should you have a need to place components or change the colour of wires, you can employ the Inspector panel. With a component selected, use the up and down triangles in the Placement section of the Inspector panel to adjust its x and y position. Meanwhile, wires can be recoloured by left-clicking to select and then changing the option in the colour box, under Properties.

Special offer for readers in **North America**



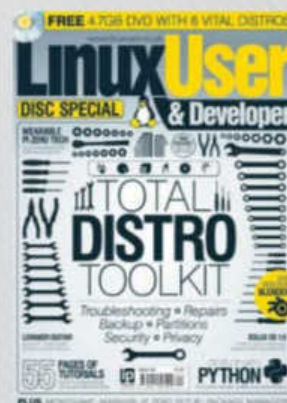
6 issues FREE

When you subscribe*

FREE
resource
downloads
in every
issue



*“The open source
authority for
professionals
and developers”*



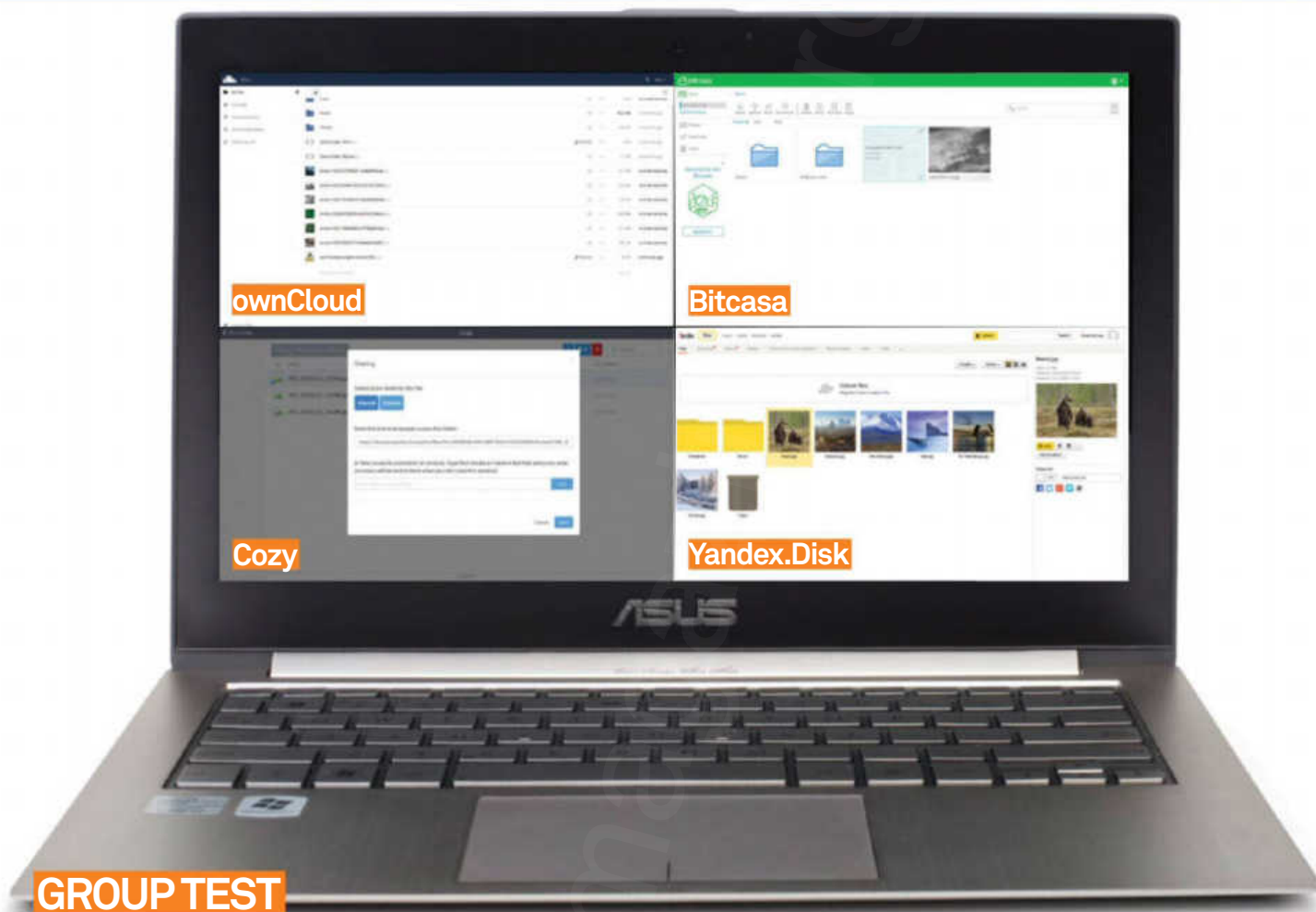
Order hotline **+44 (0)1795 418661**
Online at **www.imaginesubs.co.uk/lud**

***Terms and conditions** This is a US subscription offer. You will actually be charged £80 sterling for an annual subscription. This is equivalent to \$120 at the time of writing – exchange rate may vary. 6 free issues refers to the USA newsstand price of \$16.99 for 13 issues being \$220.87, compared with \$120 for a subscription. Your subscription starts from the next available issue and will run for 13 issues. This offer expires 31 July 2016.

Quote
USA
for this
exclusive
offer!

Reviews

26 Raspberry Pi 3 | 87 Group test | 92 Free software



GROUP TEST

Cloud storage

While Google Drive and Dropbox dominate the market in other computing areas, what cloud storage options are out there for the Linux community?

ownCloud

There are plenty of similarities between ownCloud and Dropbox, but that hasn't stopped the dev team from providing an open source alternative to the usual proprietary cloud storage providers. Built-in music streaming and content sharing makes it one of the better storage solutions around.

Download: bit.ly/1SWK2nF

Bitcasa

While Bitcasa isn't directly open source, it does use software based around GCC/Clang, OpenSSL, Rapid JSON and libCurl. It's touted as being one of the leading providers in file encryption, providing you with a virtual fortress for your various files and folders that you want to bring to the cloud.

Download: bit.ly/1p7DXIx

Cozy

Cozy provides flexibility when it comes to using it as a storage server, with additional Cozy-based apps available to considerably extend what it can do. It provides an all-in-one package for storing every file you own, and Cozy Light is also available for those accessing Cozy via a Raspberry Pi.

Download: bit.ly/1LeU29k

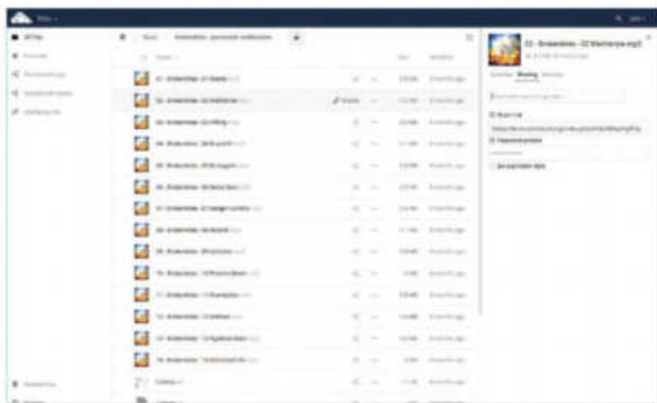
Yandex.Disk

A big part of why Yandex.Disk has gained a quite strong Linux following is likely down to its syncing system between multiple devices. There's included WebDAV support for easy management of files and a built-in flash player that enables users to preview songs before downloading them.

Download: bit.ly/1p7HSFn

ownCloud

Cloud storage built by the community, for the community



■ Use the gallery options to switch up how your files are viewed through your account

File management

One of the great things about ownCloud is that you can manage your files in different ways. There are no real limits to the folders and files you can create, and there's even support for editing web-based MS Word documents. Editing options are a little on the basic side, but it's one of the few open source storage clients that supports this.

Ease-of-use

It's not the easiest to get working straight away. Users can set up their own ownCloud server, go completely to the cloud or link their account to a domain provider. Despite it being a bit intimidating to set up, the client is fairly intuitive. It could do with a better drag-and-drop system, but we're nitpicking.

Sync and sharing

OwnCloud's core is centred around the idea of syncing and sharing files through its servers. Syncing data between accounts, and accessing it on either the desktop, web client or mobile app, is entirely possible. Sharing is a fairly standard affair, but there are additional options available for users to password-protect anything they share, or set a time limit for availability.

Extra features

Extra features are plentiful in ownCloud, including a full calendar and contact support. However, it's the collaborative editing system that has caught our eye. Up to five individuals can work on a project simultaneously, with any changes being summarised to other members via email. It's an extremely intuitive system, for the most part.

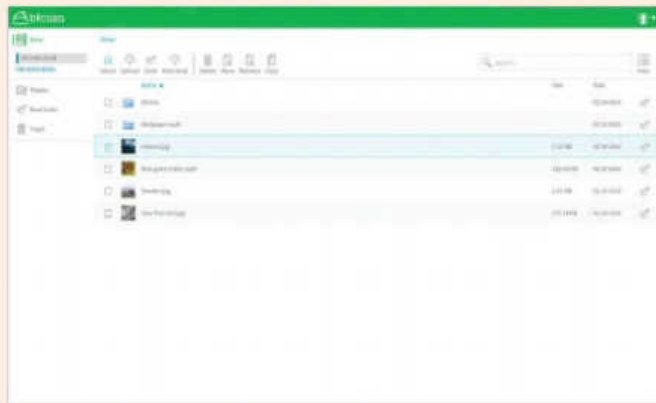
Overall

It'll take novice users some time to fully set up ownCloud, but its plethora of features and collaborative options make it a serious alternative to some of the big players out there in the proprietary pantheon.

8

Bitcasa

Can cloud encryption propel Bitcasa to the top of the class?



■ Files are automatically encrypted as soon as they're uploaded to your Bitcasa account

File management

Any and all files added to your Bitcasa account are protected by AES-256 encryption, so there's some serious protection available. It copes well with a wide array of file types, no matter their original format or the OS they stemmed from. One caveat is the suspiciously slow upload speed for some files, which didn't appear to be a problem for other cloud storage systems in this test.

Ease-of-use

Bitcasa has a lot of similarities to the likes of Dropbox and Google Drive, so it's by far the easiest to use of the group, mainly due to the lack of setup required on the user side of things. Everything from creating new folders to merging files together takes a matter of seconds to complete, plus there are embedded tutorials to help you if you're ever in doubt.

Sync and sharing

Despite being heavily tailored to encrypted files, there are still numerous sharing options within Bitcasa. Files can be sent via secure links over email, or shared among social networks when needed. Each file will still have password protection, just in case it ends up in the wrong hands.

Extra features

There aren't many extras to get excited about in Bitcasa, but a helpful auto-backup is the top option of what's included. It's a surprisingly easy way to get files into your account quickly, but without setting any limits to it, it's easy enough to have lots of unwanted files taking up all of your space.

Overall

Bitcasa's simplicity makes it a tempting proposition to most users, and while its focus on encryption should be applauded, it has left some other features feeling a little bare-bones for the most part.

7

Cozy

A new take on cloud storage that works, for the most part



■ Add extra apps to your Cozy account to expand how you can interact with your files

File management

Cozy's diverse range of self-made apps that coincide with their file management system make uploading, moving and editing files an easy enough task for even the more novice of users. Users can choose to host their Cozy account on either a physical or virtual server, depending on their expertise with the subject.

Ease-of-use

While programs like Bitcasa work out of the box, Cozy does take some initial tinkering with to get it fully working. However, once it's up and running, connecting some of Cozy's arsenal of apps is easy enough and makes using your account that little bit better. Each app has its own page for even easier access.

Sync and sharing

Syncing isn't a big part of Cozy's game and, at the time of writing, there's no cross-platform tool on hand to manage files across multiple devices - although it's expected to appear within a later update. Sharing is also minimal here, but there are basic options to email files to contacts.

Extra features

There's a large web development section built into Cozy, which can be used to edit, manipulate and make wholesale changes to your account and the apps you use. It's completely possible to acquire and improve full stack development skills here. Who knew you could keep your files safe and learn something new at the same time?

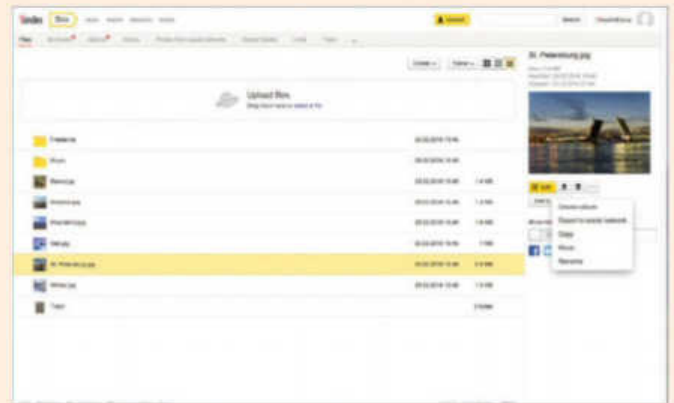
Overall

Customisation is at the heart of Cozy, and it's by far and away the best storage of choice for those who really want to make their cloud account their own without spending a fortune to do so.

9

Yandex.Disk

A multi-purpose cloud client that works across multiple platforms



■ Preview media-based files before distributing them to other areas of your account

File management

With the Linux version of Yandex.Disk consisting primarily of using a console client for moving files around, it takes more manual input than the automatic functions of the others in this group. However, it does allow for greater control of where files are placed, which will be of benefit to those who like to get a bit more hand-on with their accounts.

Ease-of-use

Going through a console client does take a little getting used to if you're coming across from a more traditional cloud storage account. But once you've mastered the basics, Yandex.Disk makes moving around the client an easy enough task that any Linux novice or pro can turn their hand to.

Sync and sharing

One of the standout features of Yandex.Disk for Linux is that you don't need to be connected to the internet to actually manage your files, so sharing between devices and contacts is a more plausible task than it is with most of the competition. Automatic syncing is another option, but it can be a little temperamental at times.

Extra features

While there's little in terms of extra features, the benefit of using a console client is that it uses minimum system resources, so it won't impair other tasks you're currently undertaking on any of its compatible distros, and it will just be much faster to use generally. For some, though, Yandex.Disk will be a little lacking in terms of what it really offers to its users.

Overall

While it does bring something different into the mix, a console client-based storage account may be a little too different for some. It's worth checking out, but only if the others don't take your fancy.

6

In brief: compare and contrast our verdicts

	ownCloud	Bitcasa	Cozy	Yandex.Disk
File management	Plenty of similarities to Dropbox, with the added bonus of built-in music streaming 9	Slow upload speeds are a bit of a bugbear in some otherwise stellar software 7	Extra applications offer new and interesting ways to manage your plethora of files 9	Using the console client isn't ideal, but it does allow for great control of files 6
Ease-of-use	Requires some legwork to get working, but the client is highly intuitive 6	Easily accessible menus and features are tailored for those seeking simplicity 8	Setting up the software and adding in the extra apps is the only bad point here 8	Once you've got used to the console client, Yandex.Disk is a breeze to use 8
Sync and share	Sharing plays a big part in ownCloud's features and it's all the better for it 9	Options aren't as vast as the competition, but cover the basics well enough for general use 7	No cross-platform support yet, but all necessary sharing options are present 7	Yandex offers an oft temperamental sharing option for distributing files 6
Extra features	Collaborative editing is the standout feature for working remotely 7	Fairly limited here, but the auto backup option can be helpful for mass uploading regularly 6	There's a built-in web development section so you can get hands-on with Cozy 8	Little in terms of extra features, but the lightweight console uses little resources 6
Overall	Solid performance. OwnCloud's varied collaborative options are of great use 8	Does a decent job of covering the basics, and you can quickly upload and move files 7	Cozy's excellent range of deep customisation options make this a formidable app 9	Worth exploring if you love console applications, but otherwise average 6

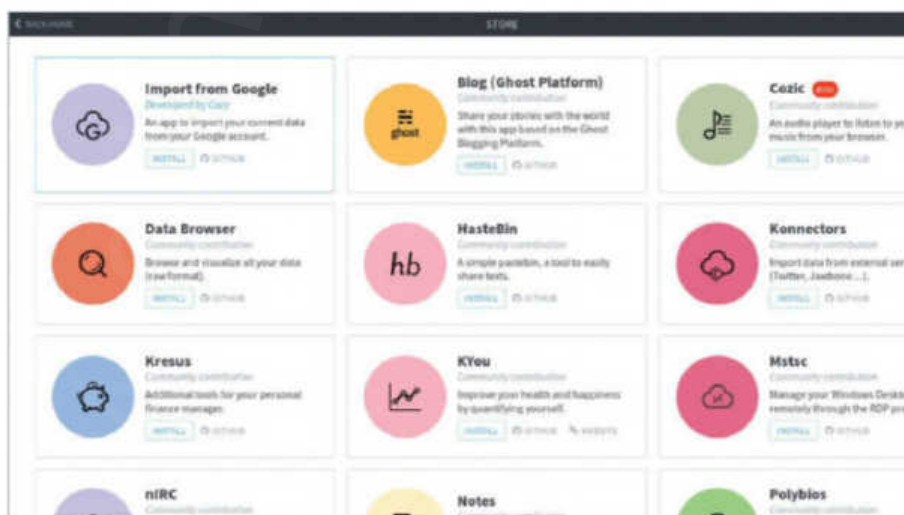
AND THE WINNER IS...



We weren't disappointed with the amount of options out there for Linux users who want to transfer their files to the cloud. No matter what your primary need for one is there's likely to be a client tailored to you, but even if you can't find one that exactly suits you, the best ones can be tinkered with behind the scenes.

The main reason we chose Cozy as our winner here is for exactly that reason: you can make it function how you need it to. As well as the staple cloud management system function, there's a whole array of additional modular apps that offer new features, alter current ones and drastically improve other areas of the software. While new users will feel at ease with the lack of manual input needed to get this set up and working, experienced users will be able to tweak and tune their experience more precisely by combining the various available applications.

Focusing on cloud storage for a moment; all the usual management options you'd expect are present and, during our time with the software, worked flawlessly. Moving files around, uploading and basic sharing are all viable options, and we especially liked its auto backup options for if manually uploading isn't



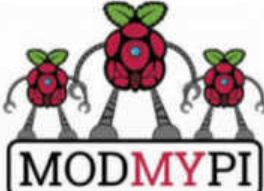
■ Cozy has an interesting array of applications that can be installed to boost functionality

for you. We should mention that the lack of cross-platform support is by far Cozy's biggest drawback currently, but it's only a drop in the ocean compared to the amount of positives offered by the service.

Eager users can currently sign up to a beta program that currently offers 60GB of storage

for free, but there are also packages available for those who want to run their own servers alongside their account; these start from a wallet-friendly \$3.49 a month. Cloud storage is at its best when it works around you – and very few can ultimately compare to what Cozy offers.

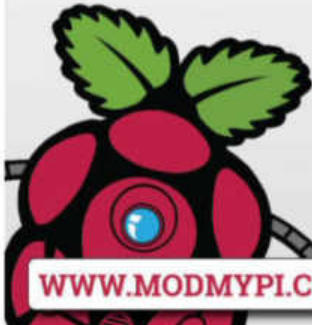
■ **Oliver Hill**



Everything Raspberry Pi!



Raspberry Pi Shop - We stock a range of breakout boards, RPi bundles, SD cards, cases, hacking kit & components - with Global Delivery!



WWW.MODMYPi.COM



HAPPY BIRTHDAY



SCAN YOUR TREAT

Celebrating TWENTY YEARS of Hosting

Come Celebrate with us and scan the QR Code to grab your birthday treat!



Domains : Hosting - Cloud - Servers

0800 808 5450



IQaudio

Audiophile accessories for the Raspberry Pi



Pi-DAC+

- Raspberry Pi HAT, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments PCM5122
- Variable output to 2.1v RMS
- Headphone Amplifier / 3.5mm socket
- Out-of-the-box Raspbian support
- Integrated hardware volume control
- Access to Raspberry Pi GPIO
- Connect to your own Hi-Fi's line-in/aux
- Industry standard Phono (RCA) sockets
- Supports the Pi-AMP+



Pi-AMP+

- Pi-DAC+ accessory, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments TPA3118
- Up to 2x35w of stereo amplification
- Provides power to the Raspberry Pi
- Software mute on GPIO22
- Auto-Mute when using Pi-DAC+ headphones
- Input voltage 12-19v
- Supports speakers from 4-8ohm



Pi-DigiAMP+

- Raspberry Pi HAT, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments TAS5756M
- Up to 2x35w of stereo amplification
- Out-of-the-box Raspbian support
- Integrated hardware volume control
- Provides power to the Raspberry Pi
- Software mute on GPIO22
- I/O (i2c, 3v, 5v, 0v, GPIO22/23/24/25)
- Just add speakers for a complete Hi-Fi
- Input voltage 12-19v
- Supports speakers from 4-8ohm



Twitter: @IQ_audio
Email: info@iqaudio.com

WWW.IQAUDIO.COM

IQaudio Limited,
Swindon, Wiltshire.
Company No.: 9461908

GAMING

SuperTux 0.4

After 11 years, the adorable SuperTux is back

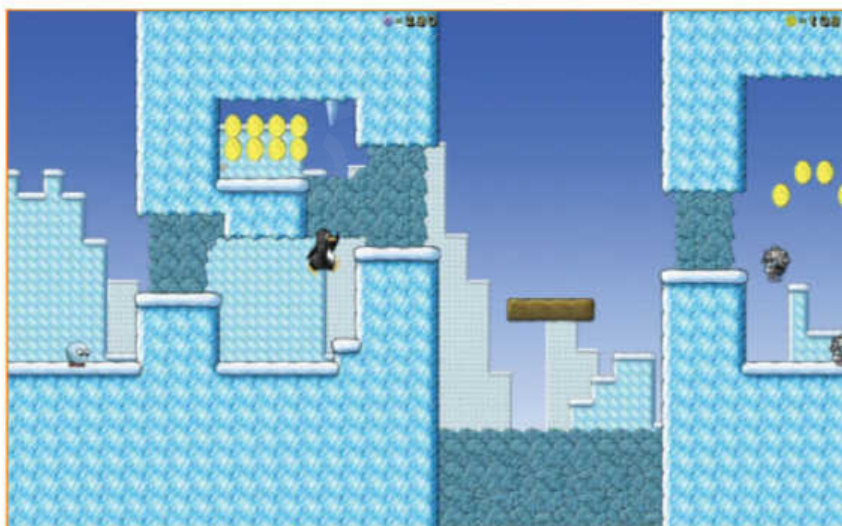


You've probably played SuperTux in the past – it's been around for a while. In your distro's repository, you'll have 0.1.3, the last stable release, dating from 2005.

Development on the unstable 3.n branch has been going along mostly unremarked for more than a decade, and now a new stable release, 0.4, bursts onto the scene!

A classic 2D sidescroller in the spirit of *Super Mario* games, originally written by Bill Kendrick of *Tux Paint* and *Circus Linux!* fame (if you have young children, those two programs are a must), *SuperTux* has undergone an extensive rewrite of the game engine and, more visibly, gameplay, graphic effects, and features. The level editor is scriptable with the Squirrel scripting language (easy to get started on if you've never used it).

We installed on LMDE with Jessie backports – with packages scheduled for Ubuntu 16.04, you may not even have to add a new repository. It's configurable for either keyboard or joystick use. Revisiting 2D side-scrollers is always fun, and time passes quickly – *SuperTux's* levels and gameplay make it every bit as addictive a challenge as navigating an Italian plumber around strange 2D worlds.



Above Admit it – you've missed playing games with cute penguins in them! Welcome back, *SuperTux*

Pros

It's fun! Not too tricky to get started with, but has its challenges. One that all of the family can enjoy.

Cons

You need to read the manual, or the bad guys will send you back to the beginning quite quickly. It's not intuitive.

Great for...

Nostalgically reliving youthful Mario Bros skills
supertuxproject.org

BACKUP

Tardis 0.26

Back up with Tardis – a time machine for your files

Backup solutions range from simple, home-brew, rsync-based scripts, to glossy commercial software, which is often just a fancy front-end to rsync-based scripts. It's essential to make backups, even if only a regular `cp -a`, because one day you *will* need to recover data. The fact that so many people don't do regular backups is in part, no doubt, owing to user inertia, but the complexity of the task, and the perceived failings of backup solutions – particularly the software's interface – takes some of the blame.

Tardis aims to be a step up from rsync-based scripts, but no more complicated than is needed. Reasonably easy to start working with, but fairly powerful and configurable, it's inspired by Time

Machine, and provides encrypted and incremental backups (and revision-like access to the archives), with filesystem-based recovery. Although it has a client/server architecture, you can run it locally – documentation is found on the homepage, and in the GitHub download.

Installation is easy on KDE desktop systems, but KDE is only a dependency for the GUI, and the project's website gives simple instructions for installing and setting up on GNOME. You need a few Python packages (Ubuntu and Fedora directions given), then you run the `setup.py` install script. If you've not made a backup this month, ask yourself why not, then try Tardis and see if it's right for you.

Pros

Local and server operation;
compact, incremental storage;
encryption; runs on the Pi

Cons

Requires some (minimal) configuration – there's no such thing as painless backup setup.

Great for...

Anyone not currently making regular backups. Do it now!
github.com/koldinger/Tardis

MULTIMEDIA

FFmpeg 3.0.0 “Einstein”

Unsung hero that makes media codecs usable for you and your apps



FFmpeg provides transcoding libraries – libavcodec, libavutil, libavformat, libavfilter, libavdevice, libswscale and libswresample – used by VLC, Xine, get-iplayer, Handbrake, and for multimedia playing within Firefox. It's also a suite of useful command line tools, including ffmpeg, a fast audio and video converter; ffplay, a simple media player; ffmpeg, a streaming server which can forward ffmpeg streams, or stream directly from disk; and ffprobe, a useful tool for gathering information from streams – either directly (the output is “human-readable”), or for use in monitoring scripts.

The FFmpeg project is dedicated to giving users the best encoders – shipping external encoder libraries

where their own are not yet mature, and offering internal and external where there is no obvious “best choice”. Until recently ffmpeg has included the VisualOn AAC encoder (libvo-aacenc), and libaacplus, but with FFmpeg 3.0 the native AAC encoder is no longer considered experimental, and the external libraries have been removed. Other changes include more HD decoders, and 30 new filters.

Installation is easy – there's no need to run the latest distributions, there's even a PPA for Ubuntu 14.04 LTS. And if 3.0.0 hasn't made it into your distro's repositories, there are 32-bit and 64-bit binaries for kernel 2.6.32 and above at <http://johnvansickle.com/ffmpeg> – but the Git builds will be more recent.

Pros

Used directly, or in scripts, you get quick and powerful results without mouse fatigue.

Cons

There are many command line switches to choose from to get the best options.

Great for...

Command line control of your video transcoding
ffmpeg.org

FEED READER

QuiteRSS 0.18.3

A cross-platform reader to tempt you from SaaS

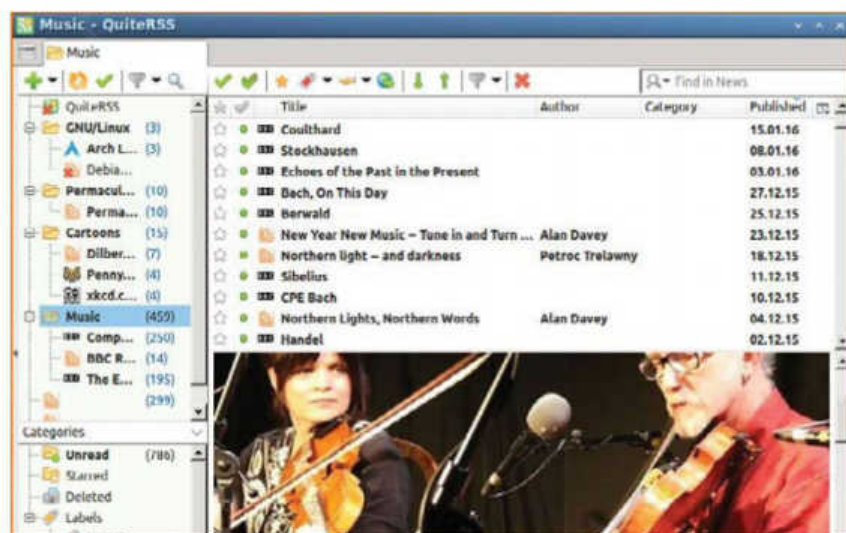


The little symbol indicating a feed of headlines from a site is perhaps not so ubiquitous as it was a few years ago, but for most important news sites there is a feed – even if it's easier to find it through Google.

Google's own Reader service closed three years ago – a small boost to desktop feed readers – but many just jumped ship to other services like Feedly.

If you'd like to bring feed reading back to your desktop, to avoid relying on some web-based service that could disappear as easily as Google Reader, QuiteRSS is a neat little reader to try. Packages are available for Arch, Fedora, FreeBSD, Gentoo, Mandriva and OpenSUSE, as well as OS X and Windows – with the pre-Qt5 version from last year also available ported to IBM's OS/2, for the seriously retro amongst you.

We did have a few issues with stability on OpenJDK 7 on Ubuntu Trusty, but we cannot blame QuiteRSS here, as this combination has given us trouble elsewhere. In daily use it was responsive enough, but most noteworthy for a simple and pleasant interface, which gave us the news we wanted, from RSS and Atom feeds, viewable in an embedded WebKit browser – with ad blocking.



Above With a hideable tree view of folders, embedded browser and filtering, get the news you want

Pros

Performant, easy-to-use, multi-lingual, and cross-platform. Easy to read, search, bookmark and filter.

Cons

Some stability issues on Ubuntu LTS with OpenJDK. Lacks the social features of some online services.

Great for...

Taking back control of RSS reading on all platforms
quiterss.org



RECOMMENDED

Hosting listings

Featured host:

www.bhost.net
sales@bhost.net



BHOST

About BHost

BHost specialises in one thing and doing it right: Linux Virtual Private Servers (VPS). BHost doesn't sell extras like domain names or SSL certificates – they are simply dedicated to providing customers with the highest quality VPS service with exceptional uptime and competitive pricing.

BHost's customer focus means that their team always goes above and beyond to help.

Their platform successfully hosts a whole variety of services, including game servers, office file storage servers, email and DNS servers, plus many thousands of websites.

BHost is confident that you will love their service. However, if for any reason you're not completely satisfied, they will provide you with a full refund within 14 days of ordering, guaranteed! Why not drop them an email?

What BHost offers

- Affordable VPS – deploy your own Linux Virtual Private Server, starting at just £5 per month
- Unlimited bandwidth allowance – you will never have to worry about data transfer quotas with BHost!
- International reach – you have a choice of data centre locations, including the UK, Netherlands and USA
- Excellent service – 100% satisfaction guaranteed, or BHost will provide you with a full refund!

5 Tips from the pros

01 Run your choice of Linux distro

A good VPS host will let you work the way you want with your preferred Linux distribution. BHost offers Ubuntu, Debian, CentOS and Fedora as standard, although if you need another distro then they will happily make it available upon request.

02 Bandwidth allowances and cost

If your server gets a surge of traffic, don't end up with a surprise bandwidth bill – look out for hosts like BHost that offer unlimited or unmetered bandwidth.

03 Port speeds – one gigabit or ten?

Your VPS will have a virtual network adaptor – check that your host allows this to run at 1-Gbit speeds, so that data transfer to and from your machine is fast.

If you're planning on hosting an application needing plenty of bandwidth, BHost can also offer 10-Gbit ports upon request.

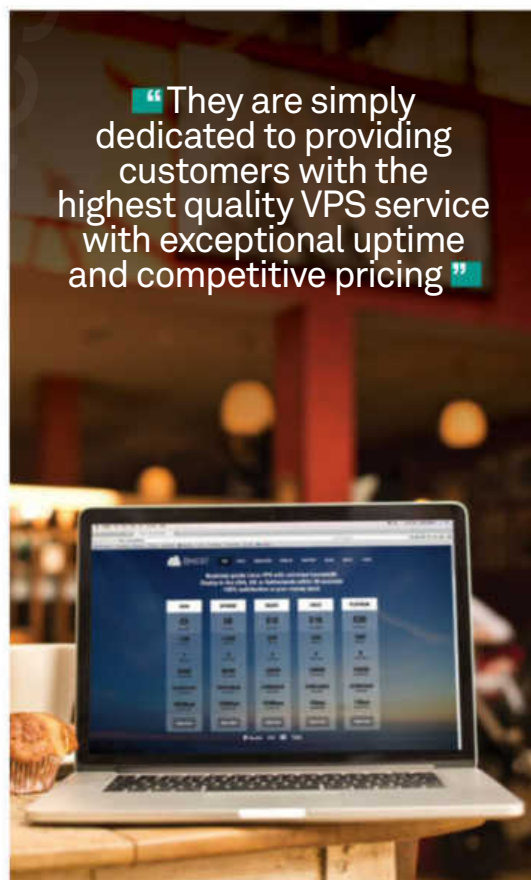
04 Build a dev machine – or ten!

Deploying a new VPS on a BHost platform takes only 30 seconds! Many of BHost's customers have several machines for development, permitting thorough testing and debugging of applications and other code before moving them to a production environment.

05 Be IPv6-ready

IPv4 addresses are running out. To ensure your services are future-proofed, find a host that fully supports the new standard: IPv6. Every BHost customer gets an IPv6-ready service as standard.

“They are simply dedicated to providing customers with the highest quality VPS service with exceptional uptime and competitive pricing”



Testimonials

Russ Murch, function28.co.uk

“I was recommended BHost as a reliable source for our hosting requirements and they have not let Function28 down. On the odd occasion when there are technical questions, BHost are very quick to respond.”

Dave Ashton, blackdoginter.net

“We have been customers of BHost for over 5 years. The facilities they offer are ideally priced for us, and BHost's service and support are absolutely second to none.”

Flamur Mavraj, @oxodesign

“After years with shared hosting, I was looking for a VPS to move my clients to. I was looking for three things: speed, reliability and low cost, and BHost have it all.”

Supreme hosting



www.cwcs.co.uk
0800 1 777 000

CWCS Managed Hosting is the UK's leading hosting specialist. They offer a fully comprehensive range of hosting products, services and support. Their 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



www.netcetera.co.uk
0800 808 5450

As the premier provider of datacentre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services to effectively manage IT infrastructures. With a state of the art data centre, Netcetera offers enterprise-level colocation and hosting.

- Managed hosting
- Domain hosting
- Datacentre and cloud

Small business host



www.hostpapa.co.uk
0800 051 7126

HostPapa is an award-winning web hosting service and a leader in green hosting. They offer 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



UK-based hosting:



www.cyberhostpro.com | 0845 5279 345

Cyber Host Pro are committed to providing the best cloud server hosting in the UK – they are obsessed with automation. They've grown year on year and love their solid, growing customer base who trust them to keep their business' cloud online! If you're looking for a

hosting provider who will provide you with the quality you need to help your business grow, then look no further than CyberHost Pro.

- Cloud VPS Servers
- Reseller hosting
- Dedicated Servers



Enterprise 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. Their team of engineers provide excellent support around the clock over the phone, email and ticketing system. ElasticHosts offers 10 data centre locations

around the world – its cloud servers can run any OS, the server specifications are adjustable by customers, and it all comes with a user-friendly web interface.

- Cloud servers on any OS
- Linux OS containers
- World-class 24/7 support

SSD Web hosting



www.bargainhost.co.uk
0843 289 2681

Since 2001 Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database which includes many repeat customers. They have 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

YOUR **FREE** RESOURCES

LOG IN TO WWW.FILESILO.CO.UK/LINUXUSER AND DOWNLOAD THE LATEST DISTROS AND FREE SOFTWARE TODAY

LATEST DISTROS

NEXT-GEN DISTROS

GET
ALL 15
DISTROS

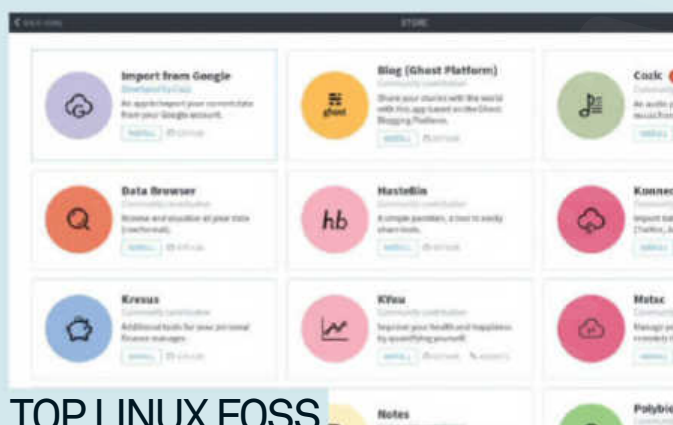
YOUR BONUS RESOURCES

ON FILESIL0 THIS ISSUE, FREE AND EXCLUSIVE FOR **LINUX USER & DEVELOPER** READERS, YOU'LL FIND THESE GREAT RESOURCES...

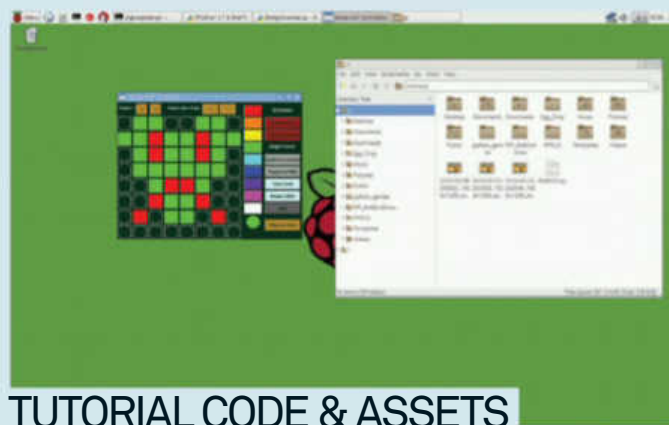
- » **15 next-gen distros**, including VeltOS, Alpine Linux, Remix OS, Qubes, NixOS and more.
- » **16 excellent FOSS packages**, including everything you'll need for this issue's tutorial line-up.
- » **Watch** The Linux Foundation, Red Hat and Raspberry Pi video guides, tutorials and webinars.
- » **Code and assets** for this issue's tutorials, including everything you need for the Raspberry Pi feature.

FileSilo

LENGTH OF VIDEO TUTORIALS: **20** HOURS



TOP LINUX FOSS



TUTORIAL CODE & ASSETS

www.filesilo.co.uk/linuxuser

FILESILO – THE HOME OF PRO RESOURCES

DISCOVER YOUR FREE ONLINE ASSETS

- A rapidly growing library
- Updated continually with cool resources
- Lets you keep your downloads organised
- Browse and access your content from anywhere
- No more torn disc pages to ruin your magazines
- No more broken discs
- Print subscribers get all the content
- Digital magazine owners get all the content too!
- Each issue's content is free with your magazine
- Secure online access to your free resources



This is the new FileSilo site that replaces your disc. You'll find it by visiting the link on the following page.

The first time you use FileSilo you'll need to register. After that, you can use the email address and password you provided to log in.

The most popular downloads are shown in this carousel, so see what your fellow readers are enjoying!

If you're looking for a particular type of content like distros or Python files, use these filters to refine your search.

Green open padlocks show the issues you have accessed. Red closed padlocks show the ones you need to buy or unlock.

Top Downloads are listed here, so you can get an instant look at the most popular downloaded content.

Check out the Highest Rated list to see the resources that other readers have voted for as the best!

Find out more about our online stores, and useful FAQs like our cookie and privacy policies and contact details.

Discover our amazing sister magazines and the wealth of content and information that they provide.

HOW TO USE

EVERYTHING YOU NEED TO KNOW ABOUT
ACCESSING YOUR NEW DIGITAL REPOSITORY



To access FileSilo, please visit www.filesilo.co.uk/linuxuser

01 Follow the instructions on-screen to create an account with our secure FileSilo system, then log in and unlock the issue by answering a simple question about the

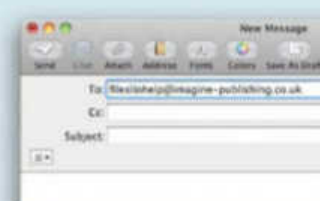


magazine. You can access the content for free with your issue.

02 If you're a print subscriber, you can easily unlock all the content by entering your unique Web ID. Your Web ID is the eight-digit alphanumeric code printed above your address details on the mailing label of your subscription copies. It can also be found on your renewal letters.

03 You can access FileSilo on any desktop, tablet or smartphone device using any popular browser (such as Firefox, Chrome or Safari). However, we recommend that you use a desktop to download content, as you may not be able to download files to your phone or tablet.

04 If you have any problems with accessing content on FileSilo, or with the registration process, take a look at the FAQs online or email filesilohelp@imagine-publishing.co.uk



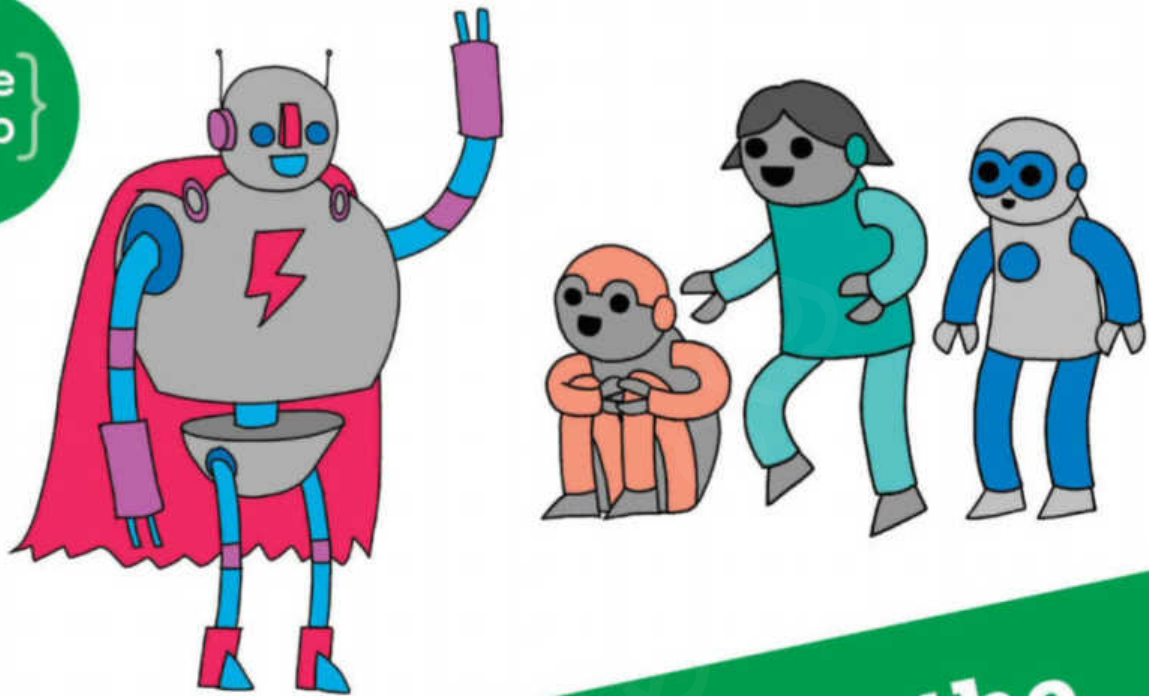
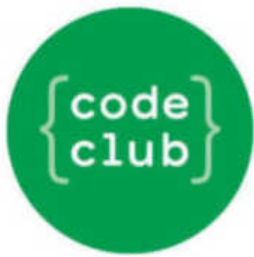
MORE TUTORIALS AND INSPIRATION

Finished reading this issue? There's plenty more free and open source goodness waiting for you on the **Linux User & Developer** website. Features, tutorials, reviews, opinion pieces and the best open source news are uploaded on a daily basis, covering Linux kernel development, the hottest new distros and FOSS, Raspberry Pi projects and interviews, programming guides and more. Join our burgeoning community of Linux users and developers and discover new Linux tools today.



www.linuxuser.co.uk

Issue 165 of **Linux User & Developer** is on sale **5 May 2016** from GreatDigitalMags.com



Can you help inspire the
next generation of coders?



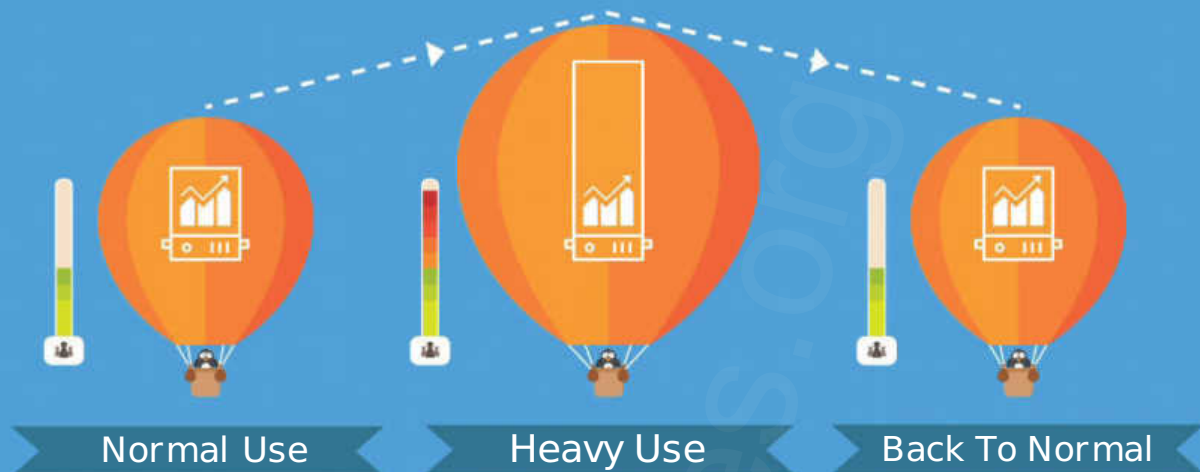
Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!
So to find out more, join us at **www.codeclub.org.uk**

elastichosts



ELASTIC CONTAINERS

- Linux-based cloud servers
- Hassle-free auto-scaling
- Unique usage-only billing

from

£3
a month

FREE TRIAL

www.elastichosts.co.uk
sales@elastichosts.com 020 7183 8250