

✓ Ubuntu ✓ Linux Mint ✓ Fedora ✓ Debian ✓ OpenSUSE

Linux

Volume 2

Tips, Tricks, Apps & Hacks

Unlock the potential of open source operating systems



Transform your system with essential software



Discover incredible distros



Manage your system efficiently



660 essential hints and tips inside

Welcome to **Linux** Tips, Tricks, Apps & Hacks

The second volume of Linux Tips, Tricks, Apps & Hacks is packed full of comprehensive features and step-by-step tutorials to help you get the most out of your Linux system. We start by looking at the best distributions of the past year, and how they stack up against one another across ten key categories. Whether you're using your Linux setup as a development platform, an entertainment system or even as an educational tool there's a distro that's suited to your needs. The Tips section that follows includes a masterclass in Python – a programming language used by beginners and experts alike – while our Tricks section features tutorials on some of the most useful free and open-source applications around and how they can improve your system. After the advanced customisation and tweaking tutorials found in the Hacks section, we review some of the best distros and apps that adhere to the FOSS philosophy.



Linux

Tips, Tricks, Apps & Hacks

Imagine Publishing Ltd
Richmond House
33 Richmond Hill
Bournemouth
Dorset BH2 6EZ
☎ +44 (0) 1202 586200

Website: www.imagine-publishing.co.uk

Twitter: @Books_Imagine

Facebook: www.facebook.com/ImagineBookazines

Head of Publishing

Aaron Asadi

Head of Design

Ross Andrews

Production Editor

Dan Collins

Senior Art Editor

Greg Whitaker

Senior Designer

Alison Innes

Printed by

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

Distributed in the UK & Eire by

Imagine Publishing Ltd, www.imagineshop.co.uk. Tel 01202 586200

Distributed in Australia by

Gordon & Gotch, Equinox Centre, 18 Rodborough Road, Frenchs Forest, NSW 2086. Tel + 61 2 9972 8800

Distributed in the Rest of the World by

Marketforce, Blue Fin Building, 110 Southwark Street, London, SE1 0SU.

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 bookazine 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 bookazine has endeavoured to ensure all information is correct at time of print, prices and availability may change. This bookazine is fully independent and not affiliated in any way with the companies mentioned herein.

Linux Tips, Tricks, Apps & Hacks Volume 2 © 2014 Imagine Publishing Ltd

ISBN 978-1909758469

Part of the

LinuxUser
& Developer
bookazine series



Contents



Tricks

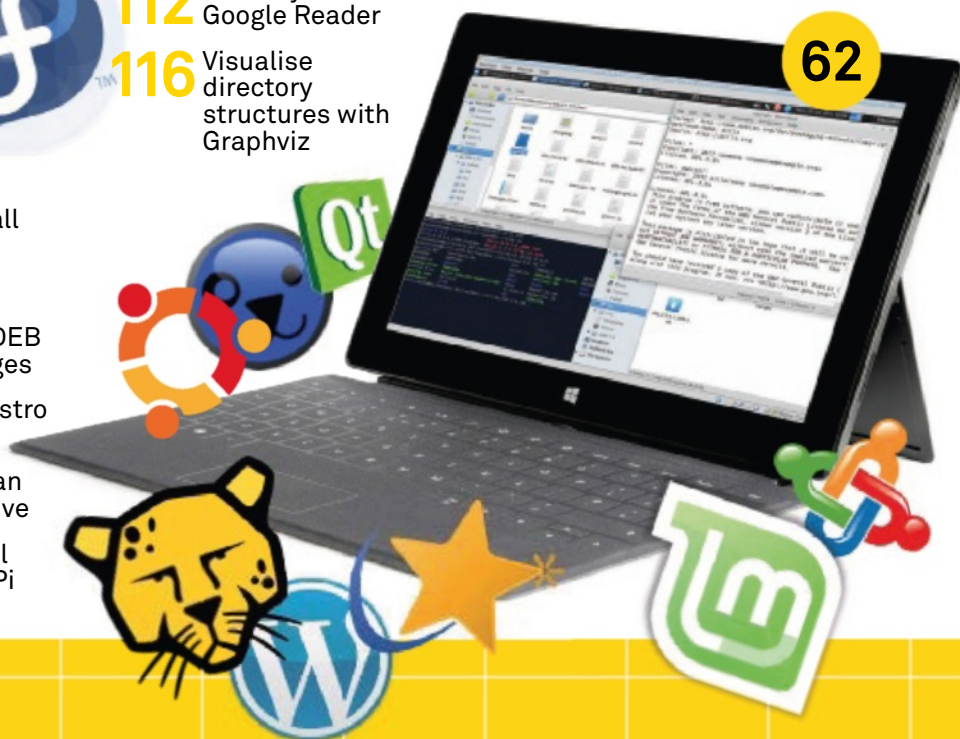
- 80 Dual boot with Windows 8
- 88 Network wirelessly with wicd
- 92 Manage your system with Webmin
- 96 Synchronise your files with Unison
- 100 Make a small business database with LibreOffice
- 104 Create and save data with a MongoDB database
- 108 Maintain and manage all of your machines with Puppet
- 112 Create your own Google Reader
- 116 Visualise directory structures with Graphviz
- 120 Create professional presentations with LaTeX
- 124 Make the most of open-source genealogy with Gramps
- 128 Take pictures and video with the Raspberry Pi Camera



“Find the best distros for your needs”

Tips

- 22 Python masterclass
- 30 Create a multi-user IM system
- 36 Get started with system administration
- 42 Test your network's security
- 46 Protect your network
- 50 Configure a secure virtual private network
- 54 Build your own pro-grade firewall
- 58 Deploy Fedora over a network
- 62 Make your own DEB and RPM packages
- 66 Install a Linux distro on Android
- 70 Dual-boot from an external hard drive
- 72 Remotely control your Raspberry Pi





Hacks

- 132** Turbocharge your cloud
- 138** Speed up Linux with Openbox
- 142** Bypass restrictive firewalls using SSH tunnelling
- 146** Create a custom build of Gentoo
- 150** Create a custom Linux kernel to optimise performance
- 154** Resize your disks on the fly with LVM
- 158** Scrape Wikipedia with BeautifulSoup
- 162** Turn your Raspberry Pi into a secure router

Apps

- | | | |
|--|----------------------------|-----------------------|
| 168 openSUSE 13.1 RC 1 | 180 Geany | 186 OpenShot |
| 170 Linux Mint 16 RC | 181 Eclipse | 187 Kdenlive |
| 172 Fedora 19 Schrödinger's Cat | 182 Parted Magic | 188 Clementine |
| 174 Mageia 3 RC | 183 System RescueCD | 189 Banshee |
| 176 AV Linux 6.0.1 | 184 Dropbox | 190 gThumb |
| 178 Bodhi Linux | 185 SpiderOak | 191 digiKam |

Ultimate distros

ULTIMATE DISTROS

We take an in-depth look at the best open-source distributions to use across ten key categories



The past year has been an interesting one for the Linux world, for one very simple reason: the Raspberry Pi. Celebrating its first birthday in 2013, the Pi has gained mainstream traction like nothing else before it and sold its two millionth unit in October.

In the world of desktop Linux, things are often relatively static: Ubuntu has sat at the top of our annual Top Ten distro lists for a long time. It took missteps from Canonical – in particular, privacy concerns regarding search and founder Mark Shuttleworth declaring that the distribution would never be truly community-driven – for that to change. But even now, it’s an Ubuntu derivative – in the form of the popular Linux Mint – that headlines.

But the Raspberry Pi has been the real shock. Over the past year, its popularity has resulted in the birth of a new distribution – Raspbian, a Debian derivative designed specifically for the Pi – which has rocketed up to take its place as one of the most widely used desktop Linux distros. Enjoying a near-monopoly in its niche – the majority of

Raspberry Pis run Raspbian, with only a few opting for alternative distributions like Raspbmc or Arch, and still fewer choosing the modern incarnation of classic ARM operating system RISC OS – Raspbian has been a surprise success story.

This isn’t the only noteworthy event of the year, though. Linus Torvalds released Linux 3.11, granting it the name ‘Linux for Workgroups’ in memory of Microsoft’s Windows for Workgroups 3.11 – the first Windows release to gain serious traction in businesses and homes. Although tongue-in-cheek, it also hints at his hopes that Linux, too, is beginning to see true traction on mainstream desktops, potentially as a response to the relatively lukewarm uptake of the latest Windows 8 release. More recently, Canonical confirmed its plans to drop X in favour of an in-house system dubbed Mir – again leaving the Ubuntu community wondering about the choices being made at the top.

Linux Mint continues to go from strength to strength, adding whizz-bang new features such as HTML5-based animated themes for the first time, a new application programming interface

for widget settings, and more. Even Fedora, after the disastrous delays that beset the launch of version 18, is back on track, launching Fedora 19 to considerable applause – and adding in a few shiny new features to help it stand out from the crowd, including an assistance system for developers.

Despite all these shake-ups, by and large the desktop Linux market is much the same now as it was 12 months ago. The popular distros continue to be popular, while the rest jostle for niche markets.

As a result, our review of the best Linux has to offer takes on a new form: rather than judging relative merits, we’ve taken ten computing categories – from general-purpose use right through to penetration testing and vintage hardware resurrection – and picked the best of the best in each individual category, along with suggestions for alternatives if you’re looking for a change.

As always, though, the choice of distribution is highly subjective. While these are our personal recommendations, the beauty of Linux is that there is a wealth of customisation available.



BEST FOR

General-purpose computing

Contenders

Linux Mint



While Linux Mint started life as a simple Ubuntu spin-off, it has overtaken Ubuntu in popularity in recent years thanks to its no-nonsense approach to design.

Ubuntu



While its attitude towards its own community may be seen as harmful, Ubuntu continues to be popular and enjoys a great deal of third-party support.

PCLinuxOS



It may be small, but PCLOS, as it is known to its fans, enjoys a great community and comes ready to use in a variety of desktop environment formats.

Fedora



The bleeding-edge Fedora offers an up-to-date environment at a small cost to stability, and suffers from less active support from commercial software developers.



■ Elegant and user-friendly, Mint is one of the best desktop distros ever made

Linux Mint

Linux Mint again reigns supreme when it comes to everyday usage



The most popular Linux distribution in the world, Linux Mint began as a spin-off from Canonical's Ubuntu – itself created from venerable Debian as a result of criticism regarding the latter's long release cycle – but has rapidly eclipsed its parent, topping the charts on download services such as DistroWatch. Available both as an Ubuntu derivative or as a Debian base, Linux Mint concentrates on sane defaults and maximising an elegant out-of-the-box experience.

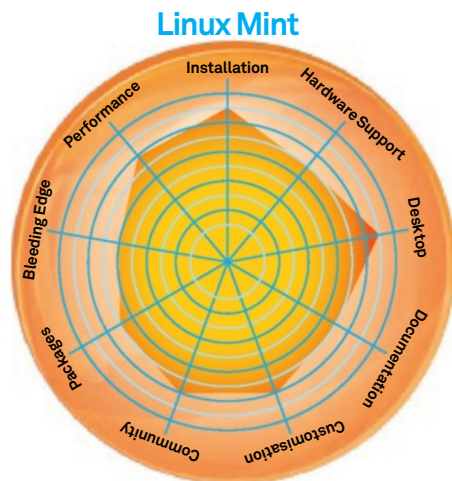
For free software purists, that can prove a problem: the distribution ships with non-free software, including Flash, Java and selected proprietary multimedia codecs, although a custom version missing the latter is available for those who are legally not permitted their use.

The primary reason for Linux Mint's success can be placed at the door of Canonical: the project was founded when the Ubuntu maintainer announced that it would be ditching support for the GNOME Shell in favour of the homebrew and extremely divisive icon-heavy Unity user interface. Forking the code base, Mint created Cinnamon and MATE – offering a traditional-looking desktop experience with



■ Mint is easy to use, even for Linux newbies

“Mint is likely to continue to grow in popularity”



the power and flexibility expected of modern environments – and saw its popularity explode when the two became standard options in Linux Mint 13 'Maya.'

Since that release, numerous bugs have been squared away to the point where we were happy, in **Linux User 128**, to give Linux Mint 15 a perfect 5/5 score, and mark it as our recommended distribution for at least the next few months.


With Canonical pushing further away from the mainstream, usurping the traditional X environment with an in-house replacement dubbed Mir, Mint is likely to continue to grow in popularity – but it's not just about being a traditional alternative: Mint is a friendly, easy-to-use distribution that's great for newcomers and professionals alike for day-to-day use.

BEST FOR

Bleeding-edge technology

Contenders


Fedora

 Fedora's rapid release cycle and penchant for testing new ground means you'll often find features that are at least six months ahead of the curve.


Arch Linux

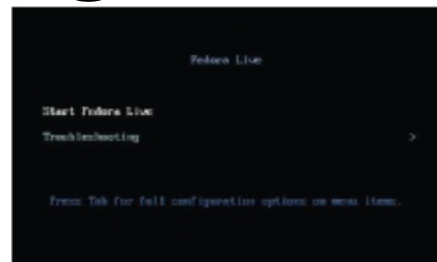
 Based on a fully rolling release cycle, Arch keeps all its users up to date, but a manual install process means it's not friendly to Linux or CLI newbies.

Gentoo Linux

 With all software compiled directly from source, rather than distributed as binaries, Gentoo is often the first with new packages and features.

Sabayon Linux

 Based on Gentoo, Sabayon attempts to meld the world of flexible rolling releases with a more user-friendly interface and approach.



It's easy to create a live CD or USB version of Fedora so you can try it on for size

Fedora

Fedora's the obvious choice for those who want the latest and greatest



Red Hat's open source offering to the world, Fedora has often been accused of the same kind of corporate meddling that saw Canonical's Ubuntu lose its place as the go-to general-purpose Linux distribution. Its fans, however, claim that such accusations are entirely untrue: while Red Hat has an overall view of its future direction, the distribution itself is driven by a meritocratic community.

The key feature of Fedora is its adherence to a rapid release cycle and a desire to be first in offering new software to its users. While this has, in the past, resulted in a less-than-polished end-user experience, more recent releases have done much to reassure users that Fedora can be trusted as a daily-use distribution – and, for those who need increased stability, there are numerous Fedora spin-offs on longer release cycles.

Those who enjoy being at the bleeding edge will find much to like in Fedora, while those who need commercial support can always get in touch with Red Hat for a contract. Although it's true that there are other distributions with even more up-to-date packages – in particular those who eschew traditional release cycles in favour of a rolling-update system – Fedora strikes an elegant balance between new features and stability, which makes it easy to recommend.

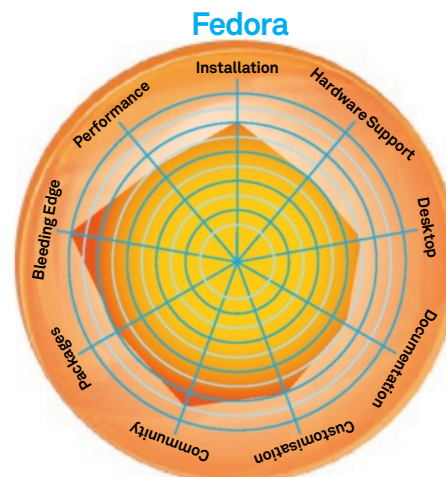
The most recent release, Fedora 19, includes the latest versions of the GNOME, Cinnamon, MATE and KDE desktop environments – although, as always, it's perfectly possible to install alternatives that are more to your taste.



Fedora offers various desktop environments, including GNOME

“An elegant balance between new features and stability”

With excellent documentation, a thriving community and the commercial backing of Red Hat, Fedora is a great choice for those who find themselves outgrowing more beginner-oriented distros – a clear sign of this is when you feel the need to start adding personal package archives (PPAs) to Ubuntu and its derivatives to work around the inclusion of older software packages.



BEST FOR

Business use

Contenders

openSUSE



One of the longest-running distributions still in active development, openSUSE boasts one of the most respected communities in FOSS.

Ubuntu



It's hard to ignore third-party support as a factor in business use, and Ubuntu remains one of the most readily supported Linux distros.

Elementary



Designed to be as easy to pick up as possible, Elementary is a good distribution for release into a Linux-hostile office environment.

Puppy



Businesses in developing nations which rely on donated or old hardware will find the performance of Puppy helps them to get things done quickly and easily.



Easy to install and user-friendly, openSUSE is a popular choice for SMEs

openSUSE

Great support and polished management tools make openSUSE a popular distro with small-to-medium enterprises

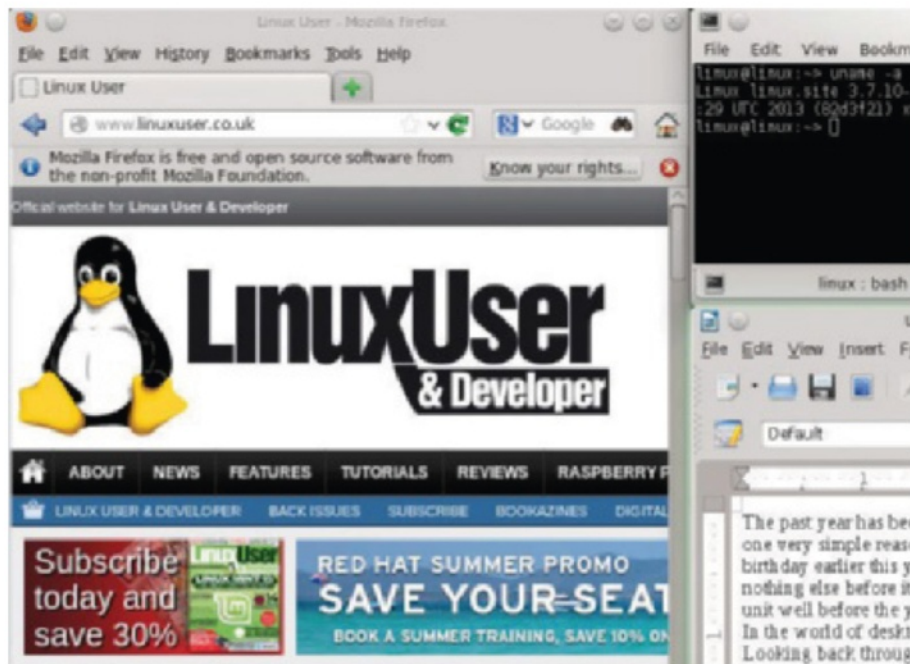


When Attachmate bought Novell back in 2011, there was concern that it would spell the end of the popular openSUSE distribution – the open source backbone of Novell's SUSE Linux Enterprise distro. Instead, Attachmate gave the project increased autonomy, spinning SUSE off from Novell as a separate entity while still offering the financial backing that any large-scale distribution requires to be truly successful.

The newest release of openSUSE, version 13.1, is the group's best yet, adhering brilliantly to the goal of providing an easy-to-use distribution with well-supported software. For companies that use SUSE's commercial products in-house, rolling openSUSE out to desktops for office use is a no-brainer: the software includes great management facilities and full access to SUSE's various commercial support services should the powers that be demand paid-for backing for a Linux roll-out.

With the distro shipping as standard with KDE and LibreOffice, the community behind openSUSE works hard to create an attractive and welcoming environment for newcomers. Although it doesn't have the same support from third-party software developers as something like Ubuntu, it has a large and thriving community and makes for a great operating system for general office use.

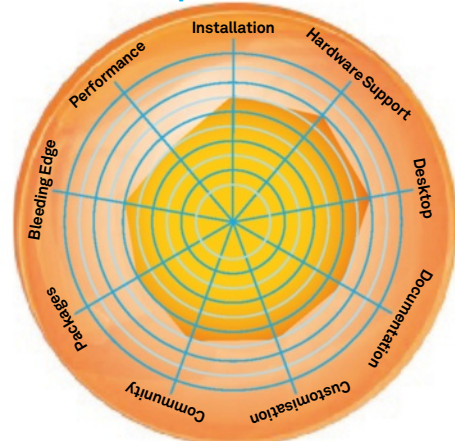
Recently, openSUSE had something of a hiccup in its release cycle which resulted in a



not-inconsiderable delay to its release, but what community manager Jos Poortvliet described as a "wake-up call" seems to have had the desired effect, with community-led changes resulting in a more streamlined process for future releases.

Perhaps the most useful feature of openSUSE is the Novell-created Open Build Service, which allows for automatic building and distribution of binary packages from source files – although the web-based service supports 22 distributions in total, of which openSUSE is just one. Each openSUSE release is built using the Open Build Service, proving its capabilities nicely.

openSUSE



BEST FOR

Development platform

Contenders

Arch Linux



Often accused as being too impenetrable to new users, Arch doesn't make things easy, but its many strengths outweigh its foibles.

CrunchBang



Also known as #!, the Openbox-based CrunchBang is designed to get out of the way and allow developers to concentrate on the task at hand.

Ubuntu



As usual, heavy support from third-party developers for Ubuntu may make its use a requirement if you rely on non-open toolchains or IDEs.

Gentoo



The aspect of building everything from source may appeal to coders who like to be able to fine-tune their environments for the best performance on any given hardware.

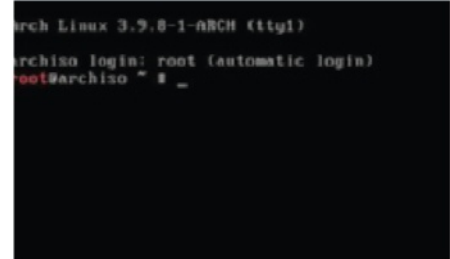
Arch Linux

It doesn't pander to beginners, but Arch is flexible and popular with kernel hackers



Arch is unapologetically not for everyone. Eschewing the modern trend of hand-holding graphical installation systems, booting the Arch installation media dumps the user straight into a console with little in the way of guidance. Consultation of the wiki provides (somewhat outdated) instructions, which leaves every step, from partitioning the drives onwards, up to the user. Even when installed, almost every package needs to be picked manually – from the desktop environment, if you decide to install one, upwards.

It's this that makes Arch popular with technical users, however: if you're willing to spend the time, or already have an in-depth knowledge of Linux, Arch gives you a fast, pared-



■ Arch isn't user-friendly, but it is very flexible

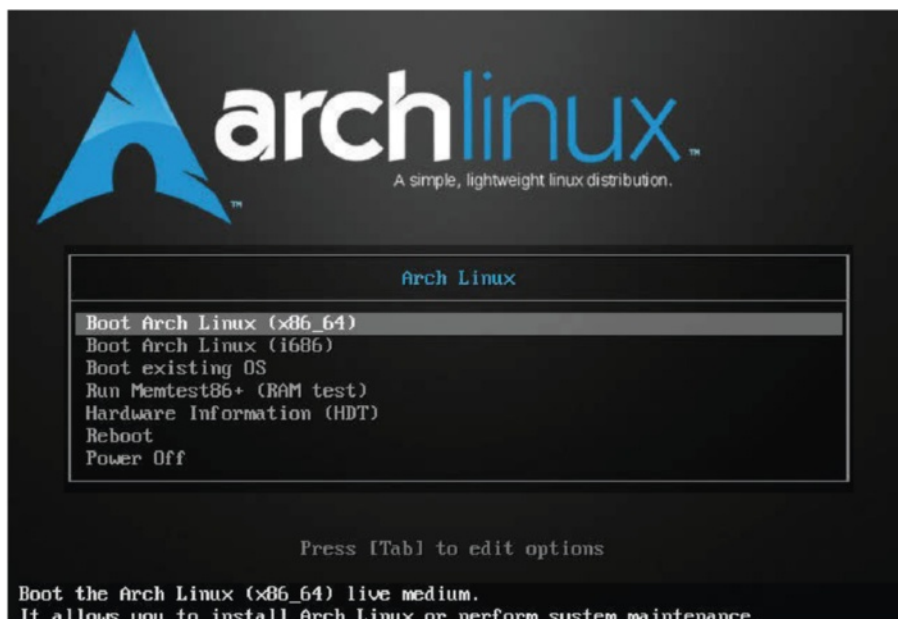
down operating system with only the packages you actually need and none of the bloat that can afflict its more beginner-oriented alternatives.

For developers, this is a particular boon – and there's little surprise that a large percentage of Arch's user base are programmers, with many even contributing to Linux development itself.

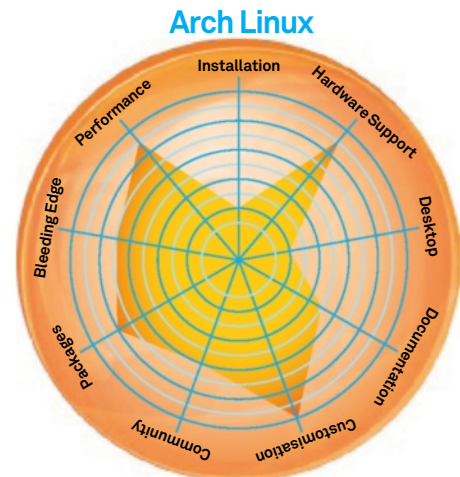
The installation process may be awkward, true, but it's also a one-off: Arch is one of the few distributions to use a true rolling-release schedule, meaning that all users automatically get the very latest packages and are always kept up to date regardless of when the initial installation was performed.

Like openSUSE, Arch also has a surprise jewel in its crown: the Arch Build System, designed to offer a platform for the creation, customisation and distribution of packages, whether independently created or simply tweaked from stock. This gives Arch a flexibility that many developers grow to depend upon, making the thought of moving to a more mainstream distribution seem like a distinct backwards step.

“Arch gives you a fast, pared-down operating system with no bloat”



■ The Arch bootloader presents the user with a range of options for boot



BEST FOR

Gaming and entertainment

Contenders

Ubuntu



Canonical's distro rose to prominence almost overnight after its initial release in 2004. Today it's the most recognised distribution in the Linux world.

Linux Mint



As an Ubuntu spin-off, most software that works in Ubuntu will also work in Linux Mint, although it lacks certain Canonical-specific features.

XBMC



For those who want a dedicated multimedia distro, XBMC provides a great 'ten-foot' user interface and support for most streaming media services.

OpenELEC



Like XBMC, OpenELEC eschews gaming in favour of a media-centric distribution that works well on a low-power system connected to a TV or projector.

Ubuntu

Canonical's deals with third-party companies keep Ubuntu on top for entertainment purposes



Ubuntu's corporate backer, Mark Shuttleworth's Canonical, comes in for a lot of flak in the open source world thanks to a somewhat poor relationship with its own community and a history of failing to contribute back as much as it takes from the wider open source world. Where Canonical shines, however, is in its relationship with other large companies – and that has directly led to Ubuntu sitting at the top of the tree as the best distribution for entertainment purposes.

The reason is simple: Canonical has partnerships with numerous other companies

to bring content to Ubuntu. Valve's recently launched Steam for Linux digital distribution platform, for example, is officially exclusive to Ubuntu – although third-party hackers have ported it to other platforms. Music-streaming service Spotify is also exclusive to parent distribution Debian, again thanks to the mainstream success enjoyed by Ubuntu and its numerous forks.

The key here, however, is the Ubuntu Software Centre. As well as acting as a front-end to the distribution's package management system, it provides a means to purchase commercial software not available on any other Linux distribution – as well as other entertainment content, including books and magazines.

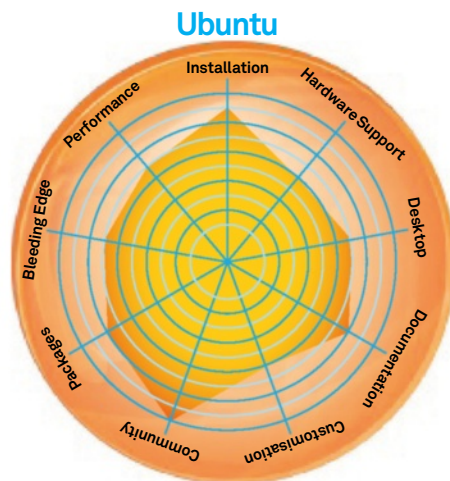
Coupled with the Unity desktop environment and its focus on providing access to web-based multimedia content – including the divisive decision to have search terms forwarded to a Canonical-owned server by default, in order to provide links to Amazon and other web shops – and Canonical's MP3 store, this makes Ubuntu the clear choice for those who like a hassle-free entertainment experience.

Ubuntu also gets bonus points in this area for the Ubuntu One service, which – for, admittedly, an extra monthly fee – provides streaming access to a user's music library on mobile devices, in addition to acting as a traditional cloud storage service.

“The clear choice for a hassle-free entertainment experience”



■ Ubuntu is easy to try out and install



■ When you search for a term, you'll also get commercially sponsored suggestions

BEST FOR

Security and penetration testing

Contenders

Kali Linux

Previously known as BackTrack Linux, Kali provides a host of utilities and tools that enable users to test the security of networks and systems.

BackBox

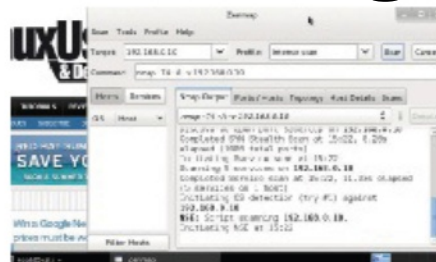
Similar to the original BackTrack (Kali's predecessor), but based on Ubuntu, BackBox is one of the most popular alternatives to Kali Linux.

Fedora Security

A 'spin' of the Fedora distribution, Fedora Security offers a great selection of penetration testing tools to test the security of your system.

Samurai Web Testing Framework

Samurai has a focus on testing web applications for security and performance issues.



■ Kali Linux offers a host of tools to test the security of your system and network

Kali Linux

BackTrack's rebirth has given the pentesting community a new queen in Kali



Formerly known as BackTrack, Kali Linux is a newcomer in its present form, launched in March. Like its predecessor, Kali aims to provide a toolkit of utilities that allow users to test their systems and networks for security leaks.

Of course, these tools could also be of use to those with less honourable intentions – and therefore Kali will always be a controversial recommendation. Its proponents point out that ne'er-do-wells will always get their hands on the tools one way or another, while its detractors complain of how easy such distributions make it to attack systems and networks – albeit as an unknowledgeable 'script kiddie.'

For those who need such tools, however, Kali comes highly recommended. Building on BackTrack, it uses a Debian Testing base – showing a focus on features over hard stability – but is designed to be suitable for daily use as a general-purpose operating system as well as a specialist live environment.

Complaints against BackTrack – such as how hard it could be to find a particular tool, with

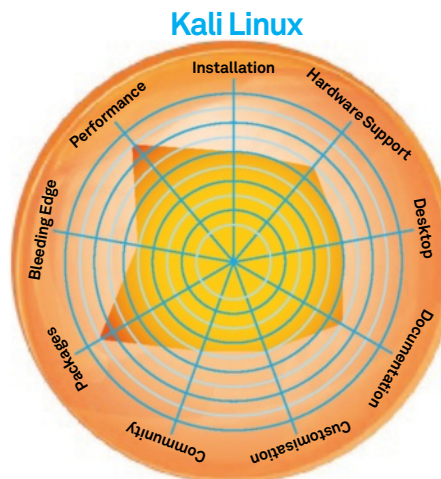
“Provides a toolkit of utilities for users to test their systems”



■ Kali's boot menu offers numerous options

everything thrown into the /pentest directory – have been taken to heart, and Kali is certainly an improvement over its predecessor. The team behind the distribution also promises improved support for ARM architecture hardware, something that was introduced in a limited form with BackTrack 5, but now – thanks to project commercial backer Offensive Security – will receive a renewed focus.

For those who need ready access to the latest penetration testing tools, Kali will be a boon. But despite improvements over BackTrack, it's still likely that it will be most commonly used as a task-specific distribution rather than a daily-use operating system for non-security-related tasks.



BEST FOR

Embedded and ARM platforms

Contenders

Raspbian

Based on Debian, Raspbian is the de facto distribution for the Raspberry Pi embedded platform and is supported by the Raspberry Pi Foundation.

Debian

A common choice for embedded platforms, Debian is available for almost any processor architecture in current production – and a few classics to boot.

Angström

Chosen for the BeagleBone and BeagleBone Black, both popular Pi alternatives, Angström targets developers with a range of neat preloaded tools.

Yocto Project

The Linux Foundation's official embedded offering for those looking to create custom distros for anything from low-power ARM to HPC-targeted Xeon chips.



Functional and clean, Raspbian works hard to be responsive

“Specifically tailored to the needs of the Raspberry Pi”

Raspbian

The Raspberry Pi's chosen operating system has gained a million users in its first year



The success of the Raspberry Pi, a low-cost ARM-based microcomputer released by the non-profit Raspberry Pi Foundation, has been incredible -- and its success in the hardware realm has been echoed by the rapid growth of a new embedded Linux distribution dubbed Raspbian.

Based on Debian, Raspbian is specifically tailored to the needs of the Raspberry Pi. It comes as standard with all the special boot files and firmwares required to get the Pi's BCM2835 processor to boot along with Pi-specific tools like raspi-config for customising the experience – including overclocking both the CPU and GPU to improve performance.

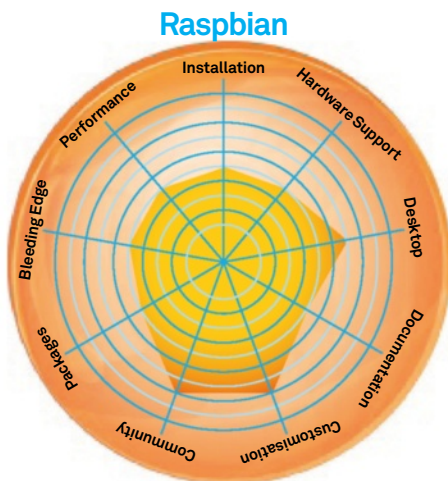
Raspbian is also seeing some traction on other embedded platforms. VIA's APC family of ARM-based computers, while shipping with Android, has a Raspbian port dubbed Apricot available. Other platforms are also looking towards Raspbian, if only to get a handle on the secret of its success. With an active user base in the millions – many of whom are technically literate hackers – it's easy to see why.



It's designed specifically for the Raspberry Pi

Raspbian's focus on the Raspberry Pi does come at a cost, however: the Pi's processor is based on the outdated ARMv6 instruction set, which means it can't take advantage of ARMv7 or ARMv8 hardware on rival single-board computers. Raspbian is also very tightly focused on the requirements of the Pi's target market – but, then again, it was ever the case for embedded distributions.


Certainly, the success of the Pi and of Raspbian has proven one thing very clearly: there is a market for low-cost, low-power hardware, and if the desktop distribution market is becoming somewhat stagnant for all but the big players, there's plenty of room for innovation and expansion in embedded development for ARM – and, if the rival processor maker has its way, MIPS – platforms.



BEST FOR

Migration from proprietary operating systems


Contenders

<p>Elementary OS</p>  <p>From the same project that brought us the popular icon sets, themes and applications of the same name, Elementary OS is really making waves.</p>	<p>Zorin OS</p>  <p>For those moving from Windows rather than Mac OS X, Zorin offers a familiar-looking environment over an Ubuntu core.</p>	<p>Netrunner</p>  <p>Based on Kubuntu, Netrunner uses a customised KDE desktop with an emphasis on friendliness to those new to Linux.</p>	<p>Ubuntu</p>  <p>While its appearance is vastly different to other platforms, the amount of help available from the Ubuntu community is not to be underestimated.</p>
---	--	--	--

“The system is designed to be familiar to users of a certain fruit-themed OS”

Elementary OS

For those coming from Apple’s OS X, Elementary OS strives to dispel the ‘Linux is hard’ myth

 In the early days of desktop Linux, usability was a serious barrier in convincing friends, family and colleagues to try the open side of the coin and install the platform alongside – or in replacement of – their existing, proprietary operating system. With most modern distributions going out of their way to be user-friendly, however, that’s no longer the case – but

for many, jumping from a proprietary OS to a new platform can still be a scary thought.

Elementary OS was developed to address exactly that issue. Based on Ubuntu, the system is designed to be familiar to users of a certain fruit-themed operating system while being easily accessible to others and with the power and flexibility one would expect of a Linux distribution.

Originally launched in 2011, the dropping of the GNOME desktop from Ubuntu led to a hiatus on the project – but it is now back with its second release, which swaps to the alternative and custom-built Pantheon desktop

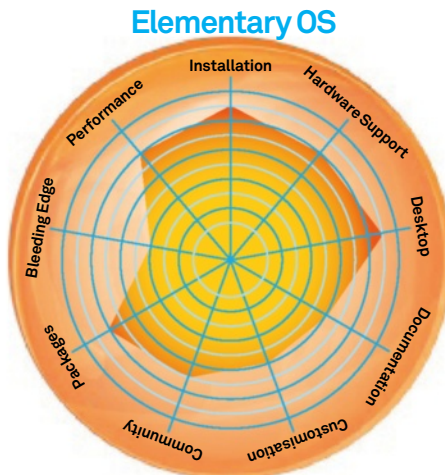
environment and includes a 64-bit release for the first time.

Interestingly, the team behind Elementary has made the choice of including a lot of custom applications with the OS in order to make things as welcoming to newcomers as possible: although third-party classics like the lightweight Midori web browser are there, the team has developed its own music player (Noise), calendar (Maya), text editor (Scratch) and others.

Sadly, this has led to a few bugs. At the time of writing, the second Elementary OS release – 0.2 ‘Luna’ – was still in beta, and there are certainly some bugs to iron out. Once it reaches its final release status, however, we predict that Elementary will become the go-to platform for those switching from at least one of the two most popular proprietary desktop operating systems.



■ Newcomers should soon feel right at home with Elementary OS



BEST FOR

Education

Contenders

UberStudent



Founded in 2010, UberStudent is a 64-bit distribution heavily focused on educational use in secondary schools and higher education.

Puppy



For establishments looking to refurbish discarded hardware, Puppy's low system requirements make it a must-have distribution.

Edubuntu



A spin-off of Ubuntu, Edubuntu includes numerous education-specific utilities and is designed specifically for non-Linux-savvy educators.

OLPC OS



Part of the One Laptop Per Child project, the Fedora-based OLPC OS offers a good interface for primary school education.

UberStudent

Putting everything a student needs at their fingertips, UberStudent makes learning almost too easy



A major area of Linux growth in recent years has been the education market, particularly in developing nations where licences for the traditional proprietary software loadout can easily eat up a school's entire budget. Easy-to-use distributions like Ubuntu excel here, but some go a step beyond and offer a pre-installed selection of education-specific packages.

UberStudent is a highlight of this breed of distro. Founded back in 2010, UberStudent is a 64-bit-only distribution – sadly excluding those trying to find a use for older hardware – designed for secondary and higher education. Based on a mixture of Debian and Ubuntu, it includes packages for both general-purpose computing – the usual loadout of media player,

web browser and office suite – as well as those targeting the education market, such as the Zotero research assistant or KeepNote note-taking application with bundled templates. An interesting addition is Artha, an open thesaurus which allows for any text to be highlighted and analysed through the WordNet database.

It's clear that Stephen Ewen, who founded and heads the project, has truly thought about the software loadout with UberStudent – although some choices, such as the inclusion of a platform for the installation of games, may raise eyebrows despite Ewen's assurances that they are designed only for use during "necessary downtime" from studies.

These small aspects aside, the latest release – UberStudent 3.0 'Plato,' launched in June this year – is a polished and well-thought-out distribution and anyone in education could do worse than give it a spin, even if only to steal some ideas for useful Linux-compatible tools and utilities to install in other distributions.

BEST FOR

Legacy or recycled computing

Contenders

Puppy



Puppy is designed specifically with legacy and mobile hardware in mind. It also runs entirely in RAM meaning fast performance and low overheads.

Damn Small Linux



Debian compatible, Damn Small Linux runs from a 50MB ISO and supports systems as old as a 486 with 16MB of RAM.

Debian



One of the oldest Linux distributions around, Debian is theoretically compatible with everything bar a 386-based system or earlier.

Tiny Core Linux



Despite weighing in at just 12MB, Tiny Core Linux includes a full graphical environment based on TinyX, FLTK and FLWM.

Puppy

It's not the most whizz-bang operating system around, but Puppy plays nicely with the oldest of PCs



Founded by Barry Kauler, Puppy Linux was never designed to compete with the big distros like Linux Mint and openSUSE. Instead, it was built – as its diminutive name suggests – as a lightweight, simple platform that allows users to get common tasks done without fuss.

As a result, Puppy is a great choice for those looking for ways to make use of outmoded and discarded hardware. Wary Puppy, the long-term support edition, is particularly well suited to this task: although it comes with an older kernel, from the 2.6 tree, it ships with the latest versions of lightweight web browsing, word processing, spreadsheet, art and email apps, along with a surprising amount of other software for its sub-140MB ISO size.

The result is that it's possible to install an attractive, user-friendly OS with useful applications on hardware that would normally be destined for landfill. For those on more modern systems, Wary Puppy's cousin Racy Puppy includes a modern kernel – and both share features like the ability to boot over a network and run on systems without local storage, or run entirely from RAM once booted in order to reduce writes on flash drives.

Even for those who don't have an old Pentium PC lying around, Puppy is worth a look. Its lightweight nature means it excels as an emergency boot disc, while those who need wider compatibility – at the cost of increased system requirements – can choose from Slackware- and Ubuntu-compatible variants in place of the homebrew Wary and Racy versions.

For those still relying on a Windows system, Puppy makes a good choice for dual-booting, with a Windows installation application and its ability to run from very limited storage space.

A look ahead to the future

We've looked at the best distros at the moment, but what about the future?

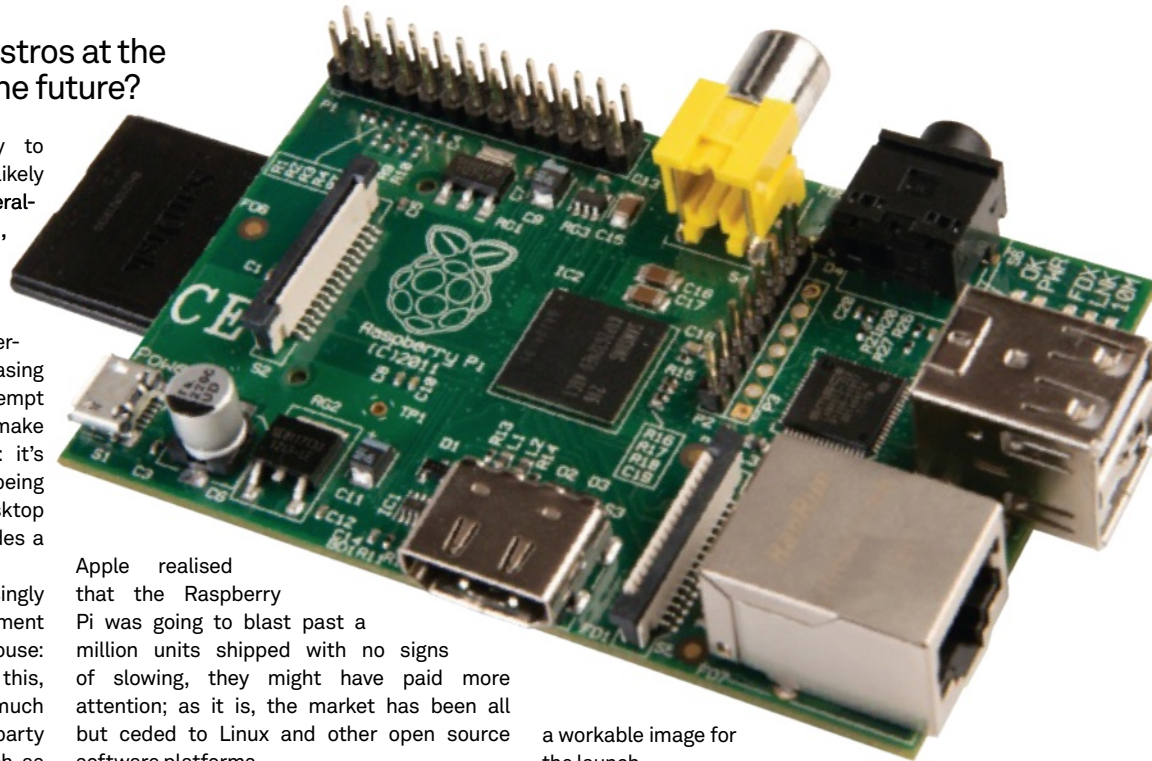
Short-term, things are easy to predict: Linux Mint will likely remain the most popular general-purpose distribution around, and Raspbian will gain users at around the same rate as the Raspberry Pi ships.

The market holds clues for longer-term trends, however. There's an increasing trend for the major distributions to attempt to gain unique selling points that make themselves stand out from the crowd: it's now less of a case of a distribution being chosen for something as gross as its desktop environment, and more because it includes a particular unique feature or application.

As a result, it's becoming increasingly common for distributions to take development of client-facing applications in-house: Elementary OS is a key example of this, using custom-written applications as much as possible and only relying on third-party projects for more complex software such as the web browser. Even outside this extreme scenario, the same thing is being repeated elsewhere: Mint uses in-house display management technologies, Ubuntu is making the move to the Mir display server in place of X, and Fedora has introduced a number of tools designed to appeal to the more technical user.

In the long term, this trend will likely result in each mainstream distribution becoming increasingly disparate from its rivals – and, as such, will almost certainly result in the birth of yet more distributions which aim to combine features from multiple other distributions, aiming to offer the best of all possible worlds. It's a cycle we've seen before in Linux development, and of which we are now approaching the mid-point with the explosion of new distros likely to occur within the next two to three years.

The biggest shift, however, is going to be towards low-cost computing. Here, the proprietary operating system vendors have been caught unawares: had Microsoft or



Apple realised that the Raspberry Pi was going to blast past a million units shipped with no signs of slowing, they might have paid more attention; as it is, the market has been all but ceded to Linux and other open source software platforms.

We're currently seeing the growth of single-board computers (SBCs) for the mainstream – as opposed to the traditional embedded development kits that predated the Pi. Companies including VIA, BeagleBoard, Samsung and Solid-Run are releasing new ARM-based SBCs seemingly every month, while AMD and Intel have both thrown their oars in with open hardware designs of their own, featuring x86 processors.

This represents a serious growth market for Linux, and one that seems to have caught the larger distributions on the hop: Raspbian became the standard for the Raspberry Pi by offering a customised and high-performance Debian derivative at a time when Fedora, the distribution previously chosen by the Raspberry Pi Foundation, had failed to deliver

a workable image for the launch.

Over the next few years, the SBC market will see increasing competition. As the hardware becomes more mature, the software will follow – but for the market to truly compete with mainstream power-hungry computers for general-purpose use, there needs to be collaboration. The current fragmented nature of the SBC market – where a distribution for one device won't run on another, even if they share a common processing platform – will need to be addressed, and the group that is first with a solution will likely find themselves leading the biggest growth in Linux use since X Window was developed.

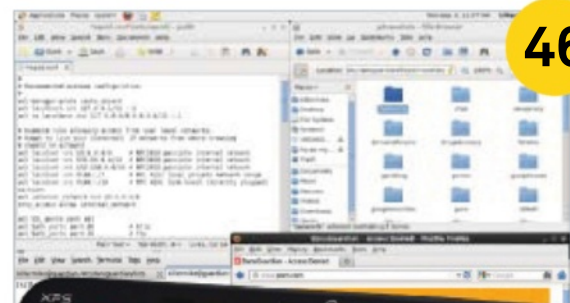
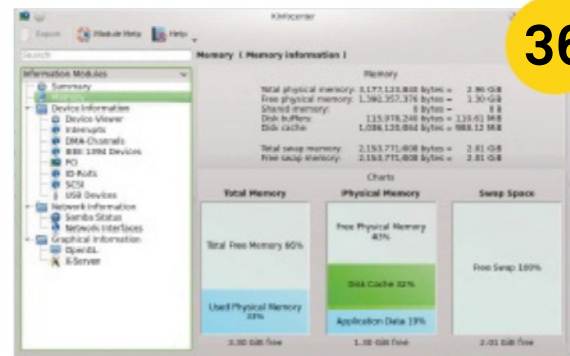
Looking into the future, one thing is abundantly clear: there has never been a better time to be involved in Linux, as a developer or simply as an end user.

“There has never been a better time to be involved in Linux”

Tips

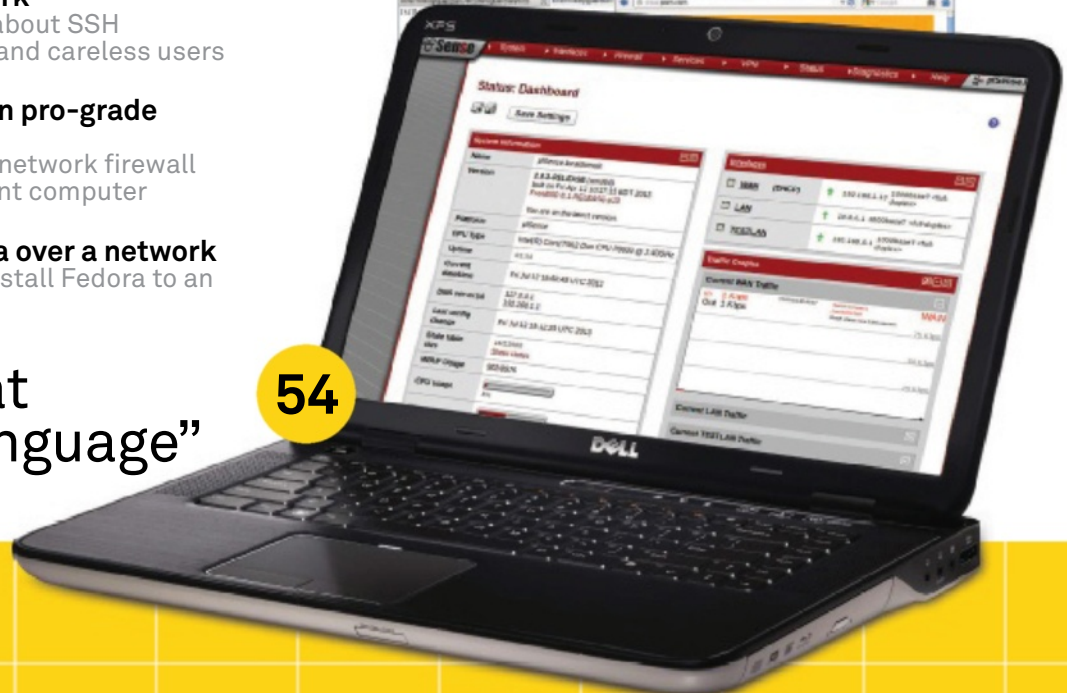
Build, create and enhance your system

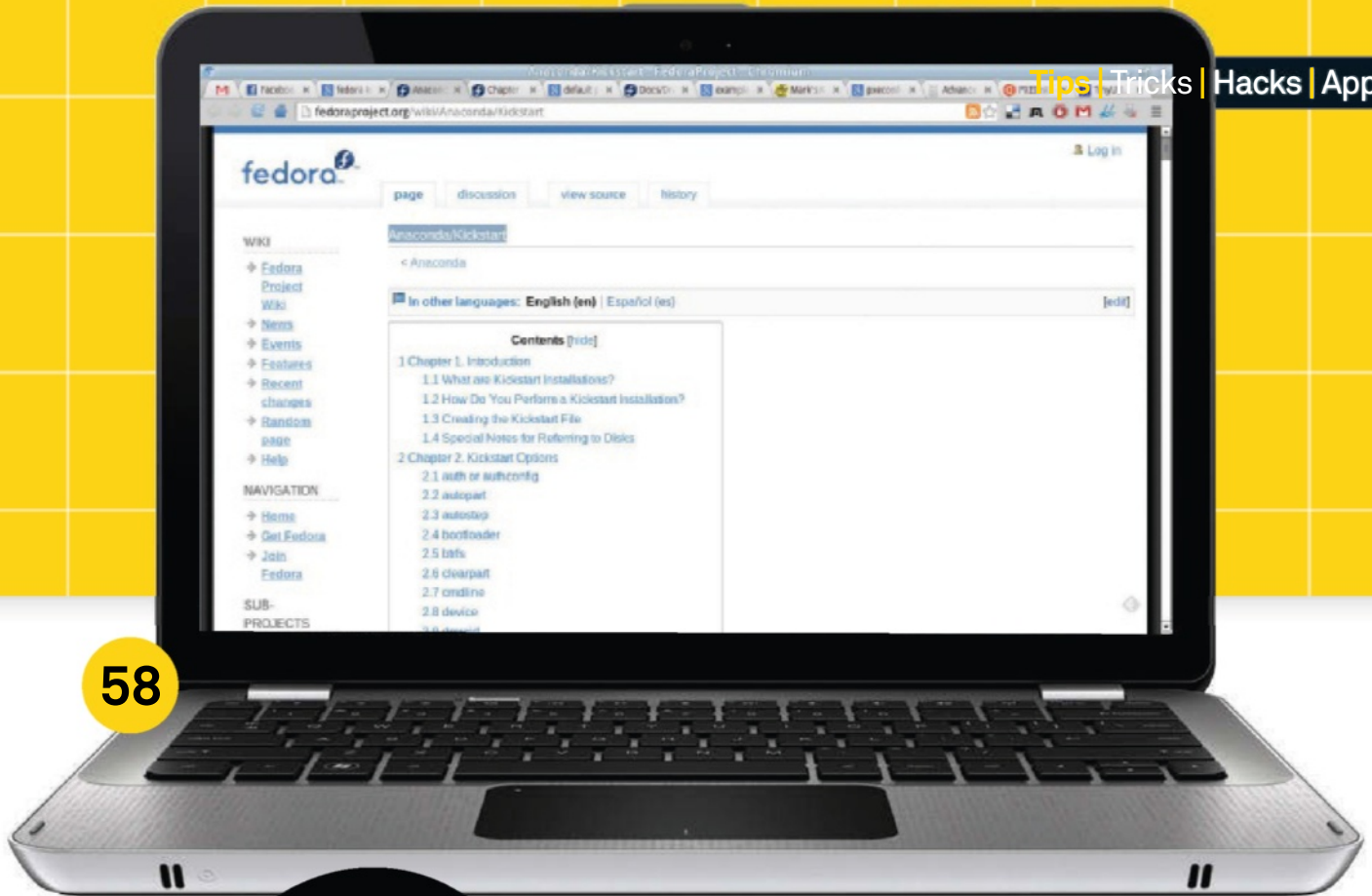
- 22** **Python masterclass**
The perfect way to get started in programming
- 30** **Create a multi-user IM system**
Program the client and server of a simple instant messenger
- 36** **Get started with system administration**
Unlock the full potential of Linux while learning how to manage it
- 42** **Test your network's security**
One of the best ways to test your security is to try to tear it apart...
- 46** **Protect your network**
Build a gateway server that can intelligently filter content
- 50** **Configure a secure virtual private network**
Stop worrying about SSH vulnerabilities and careless users
- 54** **Build your own pro-grade firewall**
Create a multi-network firewall with a redundant computer
- 58** **Deploy Fedora over a network**
Learn how to install Fedora to an entire LAN



“Python is a great programming language”

54

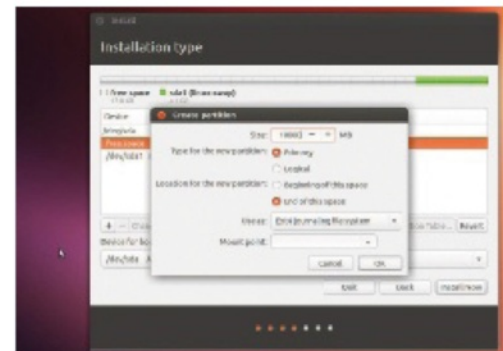




58



70



62 Make your own DEB and RPM packages

Manufacture the two most common types of Linux package

66 Install a Linux distro on Android

Keep an open source computer in your pocket

70 Dual-boot from an external hard drive

Get a multi-booting hard drive you can connect to any computer

72 Remotely control your Raspberry Pi

Take control of your Pi from your smartphone



Python Masterclass

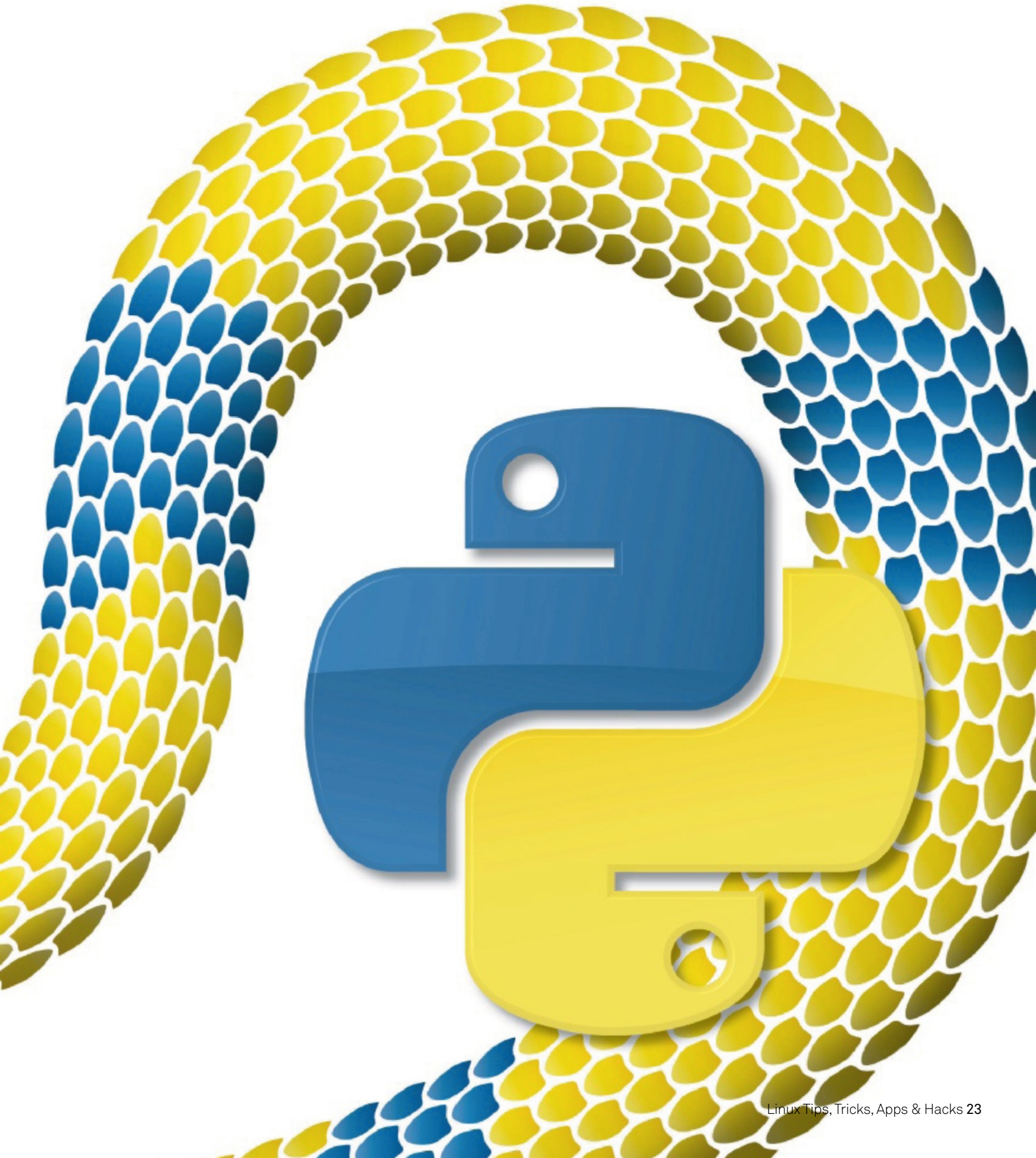
Always wanted to have a go at programming? No more excuses, because Python is the perfect way to get started!

Python is a great programming language for both beginners and experts. It is designed with code readability in mind, making it an excellent choice for beginners who are still getting used to various programming concepts. The language is popular and has plenty of libraries available, allowing programmers to get a lot done with relatively little code.

You can make all kinds of applications in Python: you could use the Pygame framework to write simple

2D games, you could use the GTK libraries to create a windowed application, or you could try something more ambitious like creating an app – in the past we've used Python's Bluetooth and Input libraries to capture the input from a USB keyboard and relay the input events an Android phone.

For this tutorial we're going to be using Python 2.x since that is the version most likely to be installed on your Linux distribution.



TIP

If you were using a graphical editor such as gedit, then you would only have to do the last step of making the file executable. You should only have to mark the file as executable once. You can freely edit the file once it's executable.

Hello World

Let's get stuck in, and what better way than with the programmer's best friend, the 'Hello World' application! Start by opening a terminal. Its current working directory will be your home directory. It's probably a good idea to make a directory for the files we'll be creating in this tutorial, rather than having them loose in your home directory. You can create a directory called Python using the command `mkdir Python`. You'll then want to change into that directory using the command `cd Python`.

The next step is to create an empty file using the command 'touch' followed by the filename. Our expert used the command `touch hello_world.py`. The final and most important part of setting up the file is making it executable. This allows us to run code inside the `hello_world.py` file. We do this with the command `chmod +x hello_world.py`. Now that we have our file set up, we can go ahead and open it up in nano, or any text editor of your choice. Gedit is a great editor with syntax highlighting support that should be available on any distribution. You'll be able to install it using your package manager if you don't have it already.

```
[liam@liam-laptop ~]$ mkdir Python
[liam@liam-laptop ~]$ cd Python/
[liam@liam-laptop Python]$ touch hello_world.py
[liam@liam-laptop Python]$ chmod +x hello_world.py
[liam@liam-laptop Python]$ nano hello_world.py
```

Our Hello World program is very simple, it only needs two lines. The first line begins with a 'shebang' (the symbol `#!` – also known as a hashbang) followed by the path to the Python interpreter. The program loader uses this line to work out what the rest of the lines need to be interpreted with. If you're running this in an IDE like IDLE, you don't necessarily need to do this.

The code that is actually read by the Python interpreter is only a single line. We're passing the value Hello World to the print function by placing it in brackets immediately after we've called the print function. Hello World is enclosed in quotation marks to indicate that it is a literal value and should not be interpreted as source code. As expected, the print function in Python prints any value that gets passed to it to the console.

You can save the changes you've made to the file in nano using the key combination `Ctrl+O`, followed by Enter. Use `Ctrl+X` to exit nano.

```
#!/usr/bin/env python2
print("Hello World")
```

You can run the Hello World program by prefixing its filename with `./` – in this case you'd type `./hello_world.py`.

```
[liam@liam-laptop Python]$ ./hello_world.py
Hello World
```

TIP

Python has plenty of great online documentation. Usually the best way to find things is to simply Google them and the first result will be the official Python documentation. For example, there is a very detailed page on Python's built-in types here: docs.python.org/2/library/stdtypes.html

Variables and data types

A variable is a name in source code that is associated with an area in memory that you can use to store data, which is then called upon throughout the code. This data can be one of many types, including:

Integer	Stores whole numbers
Float	Stores decimal numbers
Boolean	Can have a value of True or False
String	Stores a collection of characters. "Hello World" is a string

As well as these main data types, there are sequence types (technically, a string is a sequence type but is so commonly used we've classed it as a main data type):

List	Contains a collection of data in a specific order
Tuple	Contains a collection immutable data in a specific order

A tuple would be used for something like a co-ordinate, containing an x and y value stored as a single variable, whereas a list is typically used to store larger collections. The data stored in a tuple is immutable because you can't change values of individual elements in a tuple. However, you can do so in a list.

It will also be useful to know about Python's dictionary type. A dictionary is a mapped data type. It stores data in key-value pairs. This means that you access values stored in the dictionary using that value's corresponding key, which is different to how you would do it with a list. In a list, you would access an element of the list using that element's index (a number representing the element's position in the list).

Let's work on a little program we can use to demonstrate how to use variables and different data types. It's worth noting at this point that you don't always have to specify data types in Python – it will generally work out the correct data type for you. Feel free to create this file in any editor you like. Everything will work just fine as long as you remember to make the file executable. We're going to call ours `variables.py` – you can find it at <http://bit.ly/1cohPuU>, along with all the other scripts we're creating in this feature.

Interpreted vs compiled languages

An interpreted language such as Python is one where the source code is converted to machine code and then executed each time the program runs. This is different from a compiled language such as C, where the source code is only converted to machine code once – the resulting machine code is then executed each time the program runs.

The following line creates an integer variable called `hello_int` with the # value of 21. Notice how it doesn't need to go in quotation marks

The same is true of Boolean values

We create a tuple in the following way

And a list in this way

You could also create the same list in the following way

We might as well create a dictionary while we're at it. Notice how we've aligned the colons below to make the code tidy

Notice that there will now be two exclamation marks when we print the element

TIP

At this point, it's worth explaining that any text in a Python file that follows a # character will be ignored by the interpreter. This is so you can write comments in your code.

```
#!/usr/bin/env python2

# We create a variable by writing the name of the variable we want followed
# by an equals sign, which is followed by the value we want to store in the
# variable. For example, the following line creates a variable called
# hello_str, containing the string Hello World.
hello_str = "Hello World"

hello_int = 21

hello_bool = True

hello_tuple = (21, 32)

hello_list = ["Hello,", "this", "is", "a", "list"]

# This list now contains 5 strings. Notice that there are no spaces
# between these strings so if you were to join them up so make a sentence
# you'd have to add a space between each element.

hello_list = list()
hello_list.append("Hello,")
hello_list.append("this")
hello_list.append("is")
hello_list.append("a")
hello_list.append("list")

# The first line creates an empty list and the following lines use the append
# function of the list type to add elements to the list. This way of using a
# list isn't really very useful when working with strings you know of in
# advance, but it can be useful when working with dynamic data such as user
# input. This list will overwrite the first list without any warning as we
# are using the same variable name as the previous list.

hello_dict = { "first_name" : "Liam",
               "last_name"  : "Fraser",
               "eye_colour" : "Blue" }

# Let's access some elements inside our collections
# We'll start by changing the value of the last string in our hello_list and
# add an exclamation mark to the end. The "list" string is the 5th element
# in the list. However, indexes in Python are zero-based, which means the
# first element has an index of 0.

print(hello_list[4])
hello_list[4] += "!"
# The above line is the same as
hello_list[4] = hello_list[4] + "!"
print(hello_list[4])
```

Remember that tuples are immutable, although we can access the elements of them like so

Let's create a sentence using the data in our hello_dict

A tidier way of doing this would be to use Python's string formatter

```
print(str(hello_tuple[0]))
# We can't change the value of those elements like we just did with the list
# Notice the use of the str function above to explicitly convert the integer
# value inside the tuple to a string before printing it.

print(hello_dict["first_name"] + " " + hello_dict["last_name"] + " has " +
      hello_dict["eye_colour"] + " eyes.")

print("{0} {1} has {2} eyes.".format(hello_dict["first_name"],
                                     hello_dict["last_name"],
                                     hello_dict["eye_colour"]))
```

Control structures

In programming, a control structure is any kind of statement that can change the path that the code execution takes. For example, a control structure that decided to end the program if a number was less than 5 would look something like this:

```
#!/usr/bin/env python2

import sys # Used for the sys.exit function

int_condition = 5

if int_condition < 6:
    sys.exit("int_condition must be >= 6")
else:
    print("int_condition was >= 6 - continuing")
```

The path the code takes will depend on the value of the integer `int_condition`. The code in the 'if' block will only be executed if the condition is true. The import statement is used to load the Python system library; the latter provides the exit function, allowing you to exit the program, printing an error message. Notice that indentation (in this case four spaces per indent) is used to indicate which statement a block of code belongs to.

'If' statements are probably the most commonly used control structures. Other control structures include:

- For statements, which allow you to iterate over items in collections, or to repeat a piece of code a certain number of times;
- While statements, a loop that continues while the condition is true.

We're going to write a program that accepts user input from the user to demonstrate how control structures work. We're calling it **construct.py**.

The 'for' loop is using a local copy of the current value, which means any changes inside the loop won't change the

Indentation in detail

As previously mentioned, the level of indentation dictates which statement a block of code belongs to. Indentation is mandatory in Python, whereas in other languages, sets of braces are used to organise code blocks. For this reason, it is essential that you use a consistent indentation style. Four spaces are typically used to represent a single level of indentation in Python. You can use tabs, but tabs are not well defined, especially if you happen to open a file in more than one editor.

list. However, the 'while' loop is directly accessing elements in the list, so you could change the list there should you want to. We'll talk more about variable scope later. The output from the above program is as follows:

```
[liam@liam-laptop Python]$ ./construct.py
How many integers? acd
You must enter an integer

[liam@liam-laptop Python]$ ./construct.py
How many integers? 3
Please enter integer 1: t
You must enter an integer
Please enter integer 1: 5
Please enter integer 2: 2
Please enter integer 3: 6
Using a for loop
5
2
6
Using a while loop
5
2
6
```

More about a Python list

A Python list is similar to an array in other languages. A list (or tuple) in Python can contain data of multiple types, which is not usually the case with arrays in other languages. For this reason, we recommend that you only store data of the same type in a list. This should almost always be the case anyway due to the nature of the way data in a list would be processed.

The number of integers we want in the list

A list to store the integers

Used to keep track of how many integers we currently have

If the above succeeds then isint will be set to true: isint = True

By now, the user has given up or we have a list filled with integers. We can loop through these in a couple of ways. The first is with a for loop

```
#!/usr/bin/env python2

# We're going to write a program that will ask the user to input an arbitrary
# number of integers, store them in a collection, and then demonstrate how the
# collection would be used with various control structures.

import sys # Used for the sys.exit function

target_int = raw_input("How many integers? ")

# By now, the variable target_int contains a string representation of
# whatever the user typed. We need to try and convert that to an integer but
# be ready to # deal with the error if it's not. Otherwise the program will
# crash.
try:
    target_int = int(target_int)
except ValueError:
    sys.exit("You must enter an integer")

ints = list()

count = 0

# Keep asking for an integer until we have the required number
while count < target_int:
    new_int = raw_input("Please enter integer {0}: ".format(count + 1))
    isint = False
    try:
        new_int = int(new_int)

    except:
        print("You must enter an integer")

    # Only carry on if we have an integer. If not, we'll loop again
    # Notice below I use ==, which is different from =. The single equals is an
    # assignment operator whereas the double equals is a comparison operator.

    if isint == True:
        # Add the integer to the collection
        ints.append(new_int)
        # Increment the count by 1
        count += 1

print("Using a for loop")
for value in ints:
    print(str(value))
```

TIP

You can define defaults for variables if you want to be able to call the function without passing any variables through at all. You do this by putting an equals sign after the variable name. For example, you can do:

```
def modify_string
(original=" Default
String")
```

```
# Or with a while loop:
print("Using a while loop")
# We already have the total above, but knowing the len function is very
# useful.
total = len(ints)
count = 0
while count < total:
    print(str(ints[count]))
    count += 1
```

Functions and variable scope

Functions are used in programming to break processes down into smaller chunks. This often makes code much easier to read. Functions can also be reusable if designed in a certain way. Functions can have variables passed to them. Variables in Python are always passed by value, which means that a copy of the variable is passed to the function that is only valid in the scope of the function. Any changes made to the original variable inside the function will be discarded.

However, functions can also return values, so this isn't an issue. Functions are defined with the keyword `def`, followed by the name of the function. Any variables that can be passed through are put in brackets following the function's name. Multiple variables are separated by commas. The names given to the variables in these brackets are the ones that they will have in the scope of the function, regardless of what the variable that's passed to the function is called. Let's see this in action.

We are now outside of the scope of the `modify_string` function, as we have reduced the level of indentation

The test string won't be changed in this code

However, we can call the function like this

```
#!/usr/bin/env python2

# Below is a function called modify_string, which accepts a variable
# that will be called original in the scope of the function. Anything
# indented with 4 spaces under the function definition is in the
# scope.
def modify_string(original):
    original += " that has been modified."
    # At the moment, only the local copy of this string has been modified

def modify_string_return(original):
    original += " that has been modified."
    # However, we can return our local copy to the caller. The function
    # ends as soon as the return statement is used, regardless of where it
    # is in the function.
    return original

test_string = "This is a test string"

modify_string(test_string)
print(test_string)

test_string = modify_string_return(test_string)
print(test_string)

# The function's return value is stored in the variable test string,
# overwriting the original and therefore changing the value that is
# printed.
```

“The variable var is defined in a wider scope than the if statement, and can still be accessed by the if statement”

The output from the program opposite is as follows:

```
[liam@liam-laptop Python]$ ./functions_and_scope.py
This is a test string
This is a test string that has been modified.
```

Scope is an important thing to get the hang of, otherwise it can get you into some bad habits. Let's write a quick program to demonstrate this. It's going to have a Boolean variable called `cont`, which will decide if a number will be assigned to a variable in an if statement. However, the variable hasn't been defined anywhere apart from in the scope of the if statement. We'll finish off by trying to print the variable.

```
#!/usr/bin/env python2
cont = False
if cont:
    var = 1234
print(var)
```

In the code above, Python will convert the integer to a string before printing it. However, it's always a good idea to explicitly convert things to strings – especially when it comes to concatenating strings together. If you try to use the `+` operator on a string and an integer, there will be an error because it's not explicitly clear what needs to happen. The `+` operator would usually add two integers together. Having said that, Python's string formatter that we demonstrated earlier is a cleaner way of doing that. Can you see the problem? `var` has only been defined in the scope of the if statement. This means that we get a very nasty error when we try to access `var`.

```
[liam@liam-laptop Python]$ ./scope.py
Traceback (most recent call last):
  File "./scope.py", line 8, in <module>
    print var
NameError: name 'var' is not defined
```

If `cont` is set to `True`, then the variable will be created and we can access it just fine. However, this is a bad way to do things. The correct way is to initialise the variable outside of the scope of the if statement.

```
#!/usr/bin/env python2

cont = False

var = 0
if cont:
    var = 1234

if var != 0:
    print(var)
```

Comparison operators

The common comparison operators available in Python include:

<	strictly less than
<=	less than or equal
>	strictly greater than
>=	greater than or equal
==	equal
!=	not equal

The variable `var` is defined in a wider scope than the if statement, and can still be accessed by the if statement. Any changes made to `var` inside the if statement are changing the variable defined in the larger scope. This example doesn't really do anything useful apart from illustrate the potential problem, but the worst-case scenario has gone from the program crashing to printing a zero. Even that doesn't happen because we've added an extra construct to test the value of `var` before printing it.

Coding style

It's worth taking a little time to talk about coding style. It's simple to write tidy code. The key is consistency. For example, you should always name your variables in the same manner. It doesn't matter if you want to use camelCase or use underscores as we have.

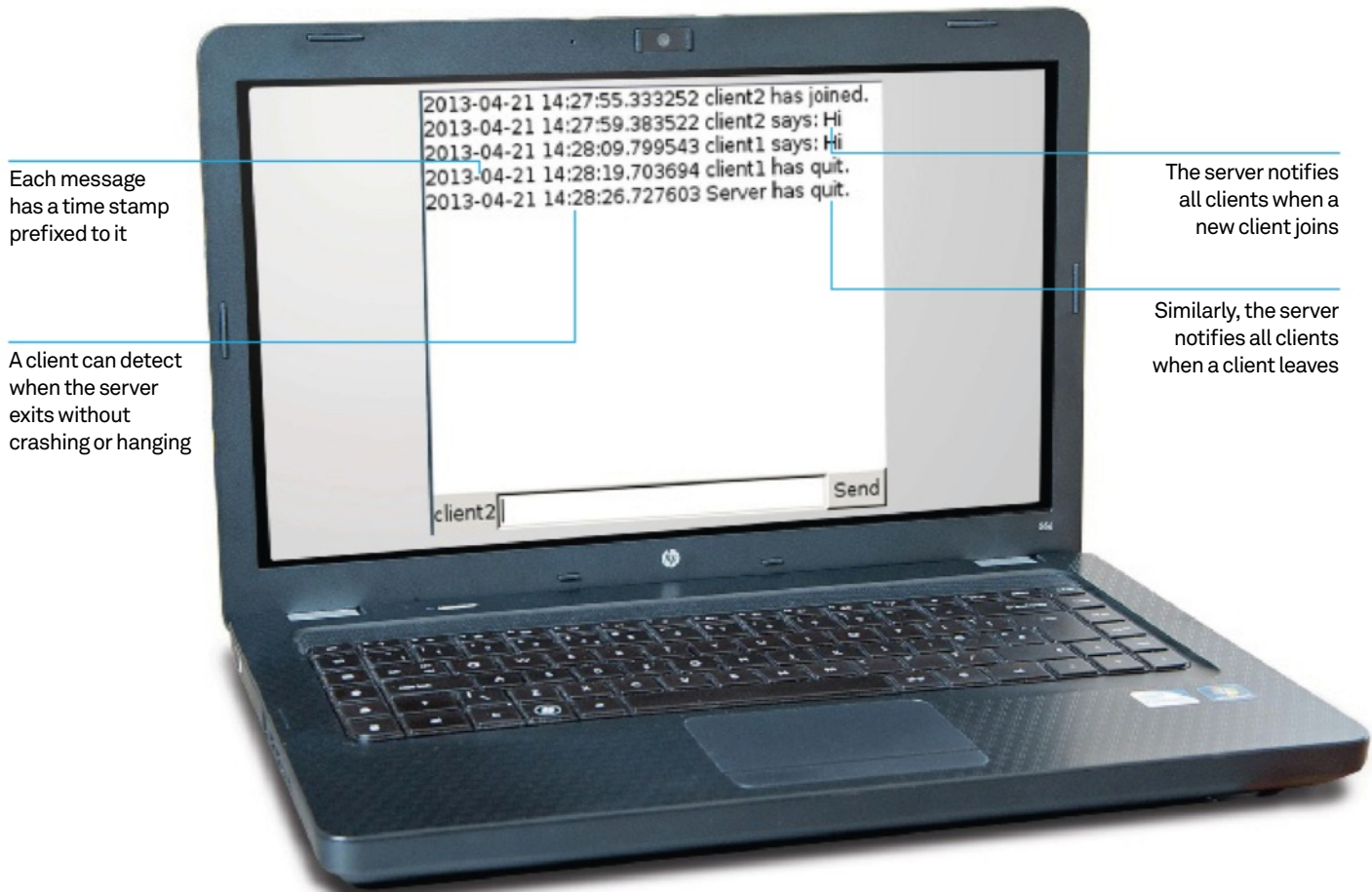
One crucial thing is to use self-documenting identifiers for variables. You shouldn't have to guess what a variable does. The other thing that goes with this is to always comment your code. This will help anyone else who reads your code, and yourself in the future. It's also useful to put a brief summary at the top of a code file describing what the application does, or a part of the application if it's made up of multiple files.

Summary

This article should have introduced you to the basics of programming in Python. Hopefully you are getting used to the syntax, indentation and general look and feel of a Python program. The next step is to learn how to come up with a problem that you want to solve, and break it down into small enough steps that you can implement in a programming language.

Google, or any other search engine, is very helpful. If you are stuck with anything, or have an error message you can't work out how to fix, stick it into Google and you should be a lot closer to solving your problem. For example, if we Google 'play mp3 file with python', the first link takes us to a Stack Overflow thread with a bunch of useful replies. Don't be afraid to get stuck in – the real fun of programming is solving problems one manageable chunk at a time.

Happy Programming!



Each message has a time stamp prefixed to it

The server notifies all clients when a new client joins

A client can detect when the server exits without crashing or hanging

Similarly, the server notifies all clients when a client leaves

Create a multi-user IM system

How to program both the client, complete with a GUI, and server of a simple instant messenger in Python

Resources

A computer – running your favourite Linux distribution

Internet connection – to access documentation

Python 2.x, PyGTK and GObject – packages installed

Today we'll be implementing an instant messenger in Python with a client-server architecture. This means each client connects to the server, which relays any message that one client sends to all other clients. The server will also notify the other clients when someone joins or leaves the server. The instant messenger can work anywhere a TCP socket can: on the same computer with the loopback interface, across various computers on a LAN, or even over the internet if you were to configure your router correctly. However, our messages aren't encrypted, so we wouldn't recommend that.

Writing an instant messenger is an interesting technical problem that covers a bunch of areas that you may not have come across while programming before:

- We'll be employing sockets, which are used to transmit data across networks.

- We'll also be using threading, which allows a program to do multiple things at once.
- We'll cover the basics of writing a simple GUI with GTK, as well as how to interact with that from a different thread.
- Finally, we'll be touching on the use of regular expressions to easily analyse and extract data from strings.

Before getting started, you'll need to have a Python2.x interpreter installed, as well as the PyGTK bindings and the Python2 GObject bindings. The chances are that you may already have these packages, so it may be easier to wait and see if you're missing any libraries when you attempt to import them. All of the above packages are commonly used, so you should be able to install them using your distro's package manager. You can download files for the tutorial from <http://bit.ly/1aqdLLa>.

01 The server

The server will do the following jobs:

- Listen for new clients
- Notify all clients when a new client joins
- Notify all clients when a client leaves
- Receive and deliver messages to all clients

We're going to write the server side of the instant messenger first, as the client requires it. There will be two code files, so it's a good idea to make a folder to keep them inside. You can create an empty file with the command `touch [filename]`, and mark that file as executable using `chmod +x [filename]`. This file is now ready to edit in your favourite editor.

```
[liam@liam-laptop Python]$ mkdir
Python-IM
[liam@liam-laptop Python]$ cd
Python-IM/
[liam@liam-laptop Python-IM]$ touch
IM-Server.py
[liam@liam-laptop Python-IM]$ chmod
+x IM-Server.py
```

02 Starting off

As usual, we need to start off with the line that tells the program loader what it needs to interpret the rest of the file with. In your advisor's case, that line is:

```
#!/usr/bin/env python2.
```

On your system, it may need to be changed to `#!/usr/bin/env python2.6` or `#!/usr/bin/env python2.7`

After that, we've written a short comment about what the application does, and imported the required libraries. We've already mentioned what the threading and socket libraries are for. The re library is used for searching strings with regular expressions. The signal library is used for dealing with signals that will kill the program, such as SIGINT. SIGINT is sent when Ctrl+C is pressed. We handle these signals so that the program can tell the clients that it's exiting rather than dying unexpectedly. The sys library is used to exit the program. Finally, the time library is used to put a sensible limit on how frequently the body of while loops execute.

```
#!/usr/bin/env python2
# The server side of an instant
messaging application. Written as
part of a Linux User & Developer
tutorial by Liam Fraser.
import threading
```

```
import socket
import re
import signal
import sys
import time
```

03 The Server class

The Server class is the main class of our instant messenger server. The initialiser of this class accepts a port number to start listening for clients on. It then creates a socket, binds the socket to the specified port on all interfaces, and then starts to listen on that port. You can optionally include an IP address in the tuple that contains the port. Passing in a blank string like we have done causes it to listen on all interfaces. The value of 1 passed to the listen function specifies the maximum number of queued connections we can accept. This shouldn't be a problem as we're not expecting a bunch of clients to connect at exactly the same time.

Now that we have a socket, we'll create an empty array that will be later used to store a collection of client sockets that we can echo messages to. The final part is to tell the signal library to run the self.signal_handler function, which we have yet to write, when a SIGINT or SIGTERM is sent to the application so that we can tidy up nicely.

```
class Server():
    def __init__(self, port):
        # Create a socket and bind it to a
        port
        self.listener = socket.
socket(socket.AF_INET, socket.SOCK_
STREAM)
        self.listener.bind(('',
port))
        self.listener.listen(1)
        print "Listening on port
{0}".format(port)
        # Used to store all of the client
sockets we have, for echoing
to them
        self.client_sockets = []
        # Run the function self.signal_
handler when Ctrl+C is pressed
        signal.signal(signal.SIGINT,
self.signal_handler)
        signal.signal(signal.
SIGTERM, self.signal_handler)
```

“You can include an IP address in the tuple that contains the port”

Useful documentation

Threading: docs.python.org/2/library/threading.html

Sockets: docs.python.org/2/library/socket.html

Regular expressions: docs.python.org/2/library/re.html

The signal handler: docs.python.org/2/library/signal.html

PyGTK: www.pygtk.org/pygtk2reference

GObject: www.pygtk.org/pygtk2reference/gobject-functions.html

04 The server's main loop

The server's main loop essentially accepts new connections from clients, adds that client's socket to the collection of sockets and then starts an instance of the ClientListener class, which we have yet to write, in a new thread. Sometimes, defining interfaces you are going to call before you've written them is good, because it can give an overview of how the program will work without worrying about the details.

Note that we're printing information as we go along, to make debugging easier should we need to do it. Sleeping at the end of the loop is useful to make sure the while loop can't run quickly enough to hang the machine. However, this is unlikely to happen as the line that accepts new connections is blocking, which means that the program waits for a connection before moving on from that line. For this reason, we need to enclose the line in a try block, so that we can catch the socket error and exit when we can no longer accept connections. This will usually be when we've closed the socket during the process of quitting the program.

```
def run(self):
    while True:
        # Listen for clients, and create a
ClientThread for each new client
        print "Listening for
more clients"
```

```

try:
    (client_socket,
client_address) = self.listener.
accept()
except socket.error:
    sys.exit("Could not
accept any more connections")

```

```

self.client_sockets.
append(client_socket)

```

```

print "Starting client
thread for {}".format(client_
address)
client_thread =
ClientListener(self, client_socket,
client_address)
client_thread.start()

```

```

time.sleep(0.1)

```

05 The echo function

We need a function that can be called from a client's thread to echo a message to each client. This function is pretty simple. The most important part is that sending data to sockets is in a try block, which means that we can handle the exception if the operation fails, rather than having the program crash.

```

def echo(self, data):
# Send a message to each socket in
self.client_socket
print "echoing: {}".
format(data)
for socket in self.client_
sockets:
# Try and echo to all clients
try:
    socket.sendall(data)
except socket.error:
    print "Unable to send
message"

```

06 Finishing the Server class

The remainder of the Server class is taken up with a couple of simple functions; one to remove a socket from the collection of sockets, which doesn't need an explanation, and the signal_handler function that we talked

about in the initialiser of the class. This function stops listening for new connections, and unbinds the socket from the port it was listening on. Finally, we send a message to each client to let them know that we are exiting. The signal will continue to close the program as expected once the signal_handler function has ended.

```

def remove_socket(self, socket):
# Remove the specified socket from the
client_sockets list
self.client_sockets.
remove(socket)
def signal_handler(self, signal,
frame):
# Run when Ctrl+C is pressed
print "Tidying up"
# Stop listening for new connections
self.listener.close()
# Let each client know we are quitting
self.echo("QUIT")

```

07 The client thread

The class that is used to deal with each client inherits the Thread class. This means that the class can be created, then started with `client_thread.start()`. At this point, the code in the run function of the class will be run in the background and the main loop of the Server class will continue to accept new connections.

We have to start by initialising the Thread base class, using the super keyword. When we created a new instance of the ClientListener class in the server's main loop, we passed through the server's self variable. We do this because it's better for each instance of the ClientListener class to have its own reference to the server, rather than using the global one that we'll create later to actually start the application.

```

class ClientListener(threading.
Thread):
def __init__(self, server,
socket, address):
# Initialise the Thread base class
super(ClientListener,
self).__init__()
# Store the values that have been
passed to the constructor
self.server = server

```

```

self.address = address
self.socket = socket
self.listening = True
self.username = "No
Username"

```

08 The client thread's loop

The loop that runs in the client thread is pretty similar to the one in the server. It keeps listening for data while self.listening is true, and passes any data it gets to a handle_msg function that we will write shortly. The value passed to the socket.recv function is the size of the buffer to use while receiving data.

```

def run(self):
# The thread's loop to receive and
process messages
while self.listening:
    data = ""
    try:
        data = self.socket.
recv(1024)
    except socket.error:
        "Unable to receive
data"
    self.handle_msg(data)
    time.sleep(0.1)
# The while loop has ended
print "Ending client thread
for {}".format(self.address)

```

09 Tidying up

We need to have a function to tidy up the thread. We'll call this either when the client sends us a blank string (indicating that it's stopped listening on the socket) or sends us the string "QUIT". When this happens, we'll echo to every client that the user has quit.

```

def quit(self):
# Tidy up and end the thread
self.listening = False
self.socket.close()
self.server.remove_
socket(self.socket)
self.server.echo("{} has
quit.\n".format(self.username))

```

10 Handling messages

There are three possible messages our clients can send:

- QUIT
- USERNAME user
- Arbitrary string to be echoed to all clients

The client will also send a bunch of empty messages if the socket has been closed, so we

“The signal will continue to close the program as expected once the signal_handler function has ended”

“One thing we need to do is to tell GObject that we’ll be using threading”

will end their thread if that happens. The code should be pretty self-explanatory apart from the regular expression part. If someone sends the USERNAME message, then the server tells every client that a new user has joined. This is tested with a regular expression. ^ indicates the start of the string, \$ indicates the end, and the brackets containing .* extract whatever comes after “USERNAME”.

```
def handle_msg(self, data):
    # Print and then process the message
    # we've just recieved
    print "{0} sent: {1}".
    format(self.address, data)
    # Use regular expressions to test for
    # a message like "USERNAME liam"
    username_result =
    re.search('^USERNAME (.*)$', data)
    if username_result:
        self.username =
        username_result.group(1)
        self.server.echo("{0}
        has joined.\n".format(self.
        username))
        elif data == "QUIT":
    # If the client has sent quit then
    # close this thread
        self.quit()
        elif data == "":
    # The socket at the other end is
    # probably closed
        self.quit()
        else:
    # It's a normal message so echo it to
    # everyone
        self.server.echo(data)
```

11 Starting the server

The code that actually starts the Server class is as follows. Note that you are probably best picking a high-numbered port as you need to be root to open ports <1024.

```
if __name__ == "__main__":
    # Start a server on port 59091
    server = Server(59091)
    server.run()
```

12 The client

Create a new file for the client as we did for the server and open it in your favourite editor. The client requires the same imports as the

server, as well as the gtk, gobject and datetime libraries. One important thing we need to do is to tell GObject that we’ll be using threading, so we can call functions from other threads and have the main window, which is running in the main GTK thread, update.

```
#!/usr/bin/env python2
# The client side of an instant
# messaging application. Written as
# part of a Linux User & Developer
# tutorial by Liam Fraser.
```

```
import threading
import gtk
import gobject
import socket
import re
import time
import datetime

# Tell gobject to expect calls from
# multiple threads
gobject.threads_init()
```

13 The client graphical user interface

The user interface of the client isn’t the main focus of the tutorial, and won’t be explained in as much detail as the rest of the code. However, the code should be fairly straightforward to read and we have provided links to documentation that will help.

Our MainWindow class inherits the gtk Window class, so we need to start by initialising that using the super keyword. Then we create the controls that will go on the window, connect any events they have to functions, and finally lay out the controls how we want. The destroy event is raised when the program is closed, and the other events should be obvious.

GTK uses a packing layout, in which you use Vboxes and Hboxes to lay out the controls. V and H stand for vertical and horizontal. These controls essentially let you split a window up almost like a table, and will automatically decide the size of the controls depending on the size of the application.

GTK doesn’t come with a control to enter basic information, such as the server’s IP address, port and your chosen username, so we’ve made a function called ask_for_info, which creates a message box, adds a text

box to it and then retrieves the results. We’ve done this because it’s simpler and uses less code than creating a new window to accept the information.

```
class MainWindow(gtk.Window):
    def __init__(self):
    # Initialise base gtk window class
        super(MainWindow, self).__
        init__()
    # Create controls
        self.set_title("IM Client")
        vbox = gtk.VBox()
        hbox = gtk.HBox()
        self.username_label = gtk.
        Label()
        self.text_entry = gtk.
        Entry()
        send_button = gtk.
        Button("Send")
        self.text_buffer = gtk.
        TextBuffer()
        text_view = gtk.
        TextView(self.text_buffer)
    # Connect events
        self.connect("destroy",
        self.graceful_quit)
        send_button.
        connect("clicked", self.send_
        message)
    # Activate event when user presses
    # Enter
        self.text_entry.
        connect("activate", self.send_
```



```

message)
# Do layout
    vbox.pack_start(text_view)
    hbox.pack_start(self.
username_label, expand = False)
    hbox.pack_start(self.text_
entry)
    hbox.pack_end(send_button,
expand = False)
    vbox.pack_end(hbox, expand
= False)
# Show ourselves
    self.add(vbox)
    self.show_all()
# Go through the configuration
process
    self.configure()
    def ask_for_info(self,
question):
# Shows a message box with a text
entry and returns the response
        dialog = gtk.
MessageDialog(parent = self, type =
gtk.MESSAGE_QUESTION,
flags = gtk.DIALOG_MODAL |

```

```

gtk.DIALOG_DESTROY_WITH_PARENT,
buttons = gtk.BUTTONS_OK_CANCEL,
message_format = question)
    entry = gtk.Entry()
    entry.show()
    dialog.vbox.pack_end(entry)
    response = dialog.run()
    response_text = entry.
get_text()
    dialog.destroy()
    if response == gtk.RESPONSE_
OK:
        return response_text
    else:
        return None

```

14 Configuring the client

This code is run after we've added the controls to the main window, and asks the user for input. Currently, the application will exit if the user enters an incorrect server address or port; but this isn't a production system, so that's fine.

```

def configure(self):
# Performs the steps to connect to
the server
# Show a dialog box asking for server
address followed by a port
    server = self.ask_for_
info("server_address:port")
# Regex that crudely matches an IP
address and a port number
    regex = re.search('^(\d+\.\
d+\.\d+\.\d+):(\d+)$', server)
    address = regex.group(1).
strip()
    port = regex.group(2).
strip()
# Ask for a username
    self.username = self.ask_
for_info("username")
    self.username_label.set_
text(self.username)
# Attempt to connect to the server
and then start listening
    self.network =
Networking(self, self.username,
address, int(port))

```

```
self.network.listen()
```

15 The remainder of MainWindow

The rest of the MainWindow class has plenty of comments to explain itself, as follows. One thing to note is that when a client sends a message, it doesn't display it in the text view straight away. The server is going to echo the message to each client, so the client simply displays its own message when the server echoes it back. This means that you can tell if the server is not receiving your messages when you don't see a message that you send.

```

def add_text(self, new_text):
# Add text to the text view
    text_with_timestamp = "{0}
{1}".format(datetime.datetime.now(),
new_text)
# Get the position of the end of
the text buffer, so we know where to
insert new text
    end_itr = self.text_buffer.
get_end_iter()
# Add new text at the end of the buffer
    self.text_buffer.insert(end_
itr, text_with_timestamp)
    def send_message(self, widget):
# Clear the text entry and send the
message to the server
# We don't need to display it as it
will be echoed back to each client,
including us.
        new_text = self.text_entry.
get_text()
        self.text_entry.set_text("")
        message = "{0} says: {1}\n".
format(self.username, new_text)
        self.network.send(message)
    def graceful_quit(self, widget):
# When the application is closed,
tell GTK to quit, then tell the
server we are quitting and tidy up
the network
        gtk.main_quit()
        self.network.send("QUIT")
        self.network.tidy_up()

```

16 The client's Networking class

Much of the client's Networking class is similar to that of the server's. One difference is

“The server is going to echo the message to each client”

that the class doesn't inherit the Thread class – we just start one of its functions as a thread.

```
class Networking():
    def __init__(self, window,
username, server, port):
# Set up the networking class
        self.window = window
        self.socket = socket.
socket(socket.AF_INET, socket.SOCK_
STREAM)
        self.socket.connect((server,
port))
        self.listening = True
# Tell the server that a new user
has joined
        self.send("USERNAME {0}".
format(username))

    def listener(self):
# A function run as a thread that
listens for new messages
        while self.listening:
            data = ""
            try:
                data = self.socket.
recv(1024)
            except socket.error:
                "Unable to receive
data"
            self.handle_msg(data)
# Don't need the while loop to be
ridiculously fast
            time.sleep(0.1)
```

17 Running a function as a thread The listener function above will be run as a thread. This is trivial to do. Enabling the daemon option on the thread means that it will die if the main thread unexpectedly ends.

```
def listen(self):
# Start the listening thread
        self.listen_thread =
threading.Thread(target=self.
listener)
# Stop the child thread from keeping
the application open
        self.listen_thread.daemon =
True
        self.listen_thread.start()
```

18 Finishing the Networking class Again, most of this code is similar to the code in the server's Networking class. One difference is that we want to add some things to the text view of our window. We do this by using the idle_add function of GObject. This allows us

to call a function that will update the window running in the main thread when it is not busy.

```
def send(self, message):
# Send a message to the server
        print "Sending: {0}".
format(message)
        try:
            self.socket.
sendall(message)
        except socket.error:
            print "Unable to send
message"

    def tidy_up(self):
# We'll be tidying up if either we are
quitting or the server is quitting
        self.listening = False
        self.socket.close()
# We won't see this if it's us
that's quitting as the window will
be gone shortly
        gobject.idle_add(self.
window.add_text, "Server has
quit.\n")

    def handle_msg(self, data):
        if data == "QUIT":
# Server is quitting
            self.tidy_up()
        elif data == "":
# Server has probably closed
unexpectedly
            self.tidy_up()
        else:
# Tell the GTK thread to add some
text when it's ready
            gobject.idle_add(self.
window.add_text, data)
```

19 Starting the client The main window is started by initialising an instance of the class. Notice that we don't need to store anything that is returned. We then start the GTK thread by calling gtk.main().

```
if __name__ == "__main__":
# Create an instance of the main
window and start the gtk main loop
    MainWindow()
    gtk.main()
```

20 Trying it out You'll want a few terminals: one to start the server, and some to run clients. Once you've started the server, open an instance of the client and enter 127.0.0.1:port, where 'port' is the port you decided to use. The server

“Enter a username and click OK”

will print the port it's listening on to make this easy. Then enter a username and click OK. Here is an example output from the server with two clients. You can use the client over a network by replacing 127.0.0.1 with the IP address of the server. You may have to let the port through your computer's firewall if it's not working.

```
[liam@liam-laptop Python]$ ./IM-
Server.py
Listening on port 59091
Listening for more clients
Starting client thread for
('127.0.0.1', 38726)
('127.0.0.1', 38726) sent: USERNAME
client1
echoing: client1 has joined.
Listening for more clients
Starting client thread for
('127.0.0.1', 38739)
('127.0.0.1', 38739) sent: USERNAME
client2
echoing: client2 has joined.
Listening for more clients
('127.0.0.1', 38739) sent: client2
says: Hi
echoing: client2 says: Hi
('127.0.0.1', 38726) sent: client1
says: Hi
echoing: client1 says: Hi
('127.0.0.1', 38726) sent: QUIT
echoing: client1 has quit.
Ending client thread for
('127.0.0.1', 38726)
^CTidying up
echoing: QUIT
Could not accept any more
connections
('127.0.0.1', 38739) sent:
echoing: client2 has quit.
Ending client thread for
('127.0.0.1', 38739)
```

21 That's it! So, it's not perfect and could be a little more robust in terms of error handling, but we have a working instant messenger server that can accept multiple clients and relay messages between them. More importantly, we have learned a bunch of new concepts and methods of working.

Get started with system administration

Unlock the full potential of Linux while learning how to manage it effectively...

Linux is the operating system that has more network card drivers than video card drivers, if you catch our drift. Linux was made for network. Granted, it's not too shabby in other areas, but it really excels in the networked environment. Today Linux powers most of the world's servers, whether on the internet or an intranet. One of the core competencies of Linux, which has made it perfect for running servers and services, is its

system administration features. These aren't just useful for servers in multimillion-pound companies, but even if you're using Linux at home. They give you a very smart and efficient way to control and optimise your system to your exact requirements. This article is designed to teach you about Linux system administration from a beginner's point of view. Most of the tasks we will cover can be carried out by readers who

are relatively new to 'getting their hands dirty', but we'll also cover a good few advanced tips for those who want to delve a little bit deeper.

“Linux powers most of the world's servers”

KEY

\$

= regular user commands

#

= root user commands

The latter must be used as root or by using the sudo command.

Advanced Tip:

If you are looking for single sign-on for the applications and services, you should look into Linux pluggable authentication modules (PAM). PAM provides a plug-in like architecture to develop authentication back-ends. There are many PAM modules in existence, such as FTP, OpenPGP smartcards etc. You can see the complete list of available modules at www.linux-pam.org/modules.html. This will save you lots of time creating individual users and your users will enjoy the freedom of using their existing credentials instead of remembering new ones.

The tools

In this section we'll look at doing things using some of the tools designed to help us in system administration tasks.

Managing users

While installing Linux you are asked to create at least two users for the system. One is root, which has the ultimate power over the system, and the other one is the regular user – restricted to performing day-to-day tasks. Let's see what else is possible with regards to users.

To add a user:

adduser <username>

On some systems (such as Ubuntu) you will also be asked enter the password for the new user. On other systems you will need to create passwords separately:

passwd <username>

The passwd command can also be used to change other users' passwords. When not used with a username, it offers to change the password for the user issuing the command.

Installing packages

Most Linux distributions use either the Debian package format (DEB) or Red Hat Package Manager (RPM). As already evident by the package format name, DEB is used on Debian-based distributions such as Ubuntu and Knoppix, while RPM is used on Red Hat

Linux-based distributions such as Fedora and openSUSE.




To install a Debian package:
dpkg -i packagename.deb



To install an RPM package:
rpm -i packagename.rpm


While the dpkg and rpm commands look pretty straightforward, they are very difficult to use practically because of dependency. Each RPM/DEB package is always dependent on some other RPM/DEB package; if you do not have the required package in the exact version number, the install will not succeed. So in order to install one package, you have to hunt down the package it depends on, then install it. By the way, you will also have hunt down the dependent packages for the packages your original package depends on.

To work around this issue, Linux distributions have created high-level package managers which automatically download the packages and resolve all of the dependencies. The only problem with this approach is it's not standard across all distros.

 On Fedora/Red Hat you can use Yellowdog Updater, Modified (YUM):

yum install <packagename>

Note: YUM can also be installed on other distributions such as Ubuntu and openSUSE.

 On Debian/Ubuntu you can use Advanced Packaging Tool (Apt):

apt-get install <packagename>

 On openSUSE you can use Zypp:

zypper install <packageName>

Managing services

In Linux, a service is a crucial application (or collection of applications) that runs in the background. They handle everything from booting the system to serving webpages. You can use the command 'service' (an init script) to manage services.

To get the status of all the services installed on the system:

service --status-all

To start a service:

service <service name> start

To stop a service:

service <service name> stop

To get the status of particular service:

service <service name> status

Running scheduled tasks

If you are doing a repetitive task on your system, it is better to automate. For example, you may want to sync files between two systems at a regular interval. Instead of doing it yourself manually, you can create a scheduled task that automatically runs at the configured intervals. In Linux (and most UNIX environments) this is achieved through cron. Cron is a time-based task scheduler.

To create a scheduled tasks using cron...

01 Run the following command to open the current user's crontab file:

\$ crontab -e

If you want a task to be run using root privileges, you should use the command:

\$ sudo crontab -e

02 The crontab file will then open in the default text editor.

The default text editor can be set up using the EDITOR environment variable:

\$ export EDITOR=nano

“If you are doing a repetitive task on your system, it is better to automate”

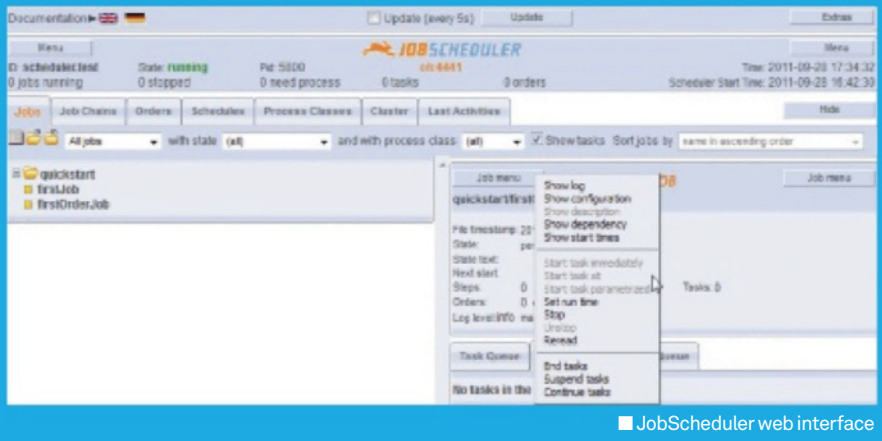
Advanced Tip:

Cron is not the only task scheduler out there. There are a number of alternatives available. One we really like is JobScheduler. It provides the following advantages over cron:

- Provides a log file for running programs.
- The execution status of a program is checked

automatically and is reported to the administrator automatically.

- You can start jobs in a sequence that is dependent on the execution status of the jobs.
- You can use a centralised user interface to manage, configure and monitor jobs.



JobScheduler web interface

Crontab takes input in the following format:

**minute(0-59) hour(0-23) day(1-31)
month(1-12) weekday(0-6) command**

An asterisk (*) is used as wild card. For example, using asterisk with month will cause the task to run every month.

03 Let's assume that you want to run /usr/bin/myludapp every day at 12.30 AM. So we will need to create the following line in it:

29 0 * * * /usr/bin/myludapp

Here, 29 is for the 30-minute mark and 0 for 12 am because the minute, hour and weekday values start at 0. However, the day and month values start at 1 instead of 0.

Managing backups

Backup is very crucial to any system, whether running in isolation or a networked environment. You can use rsync to create backups for your system. Rsync is a file synchronisation utility. It provides the following features which make it a perfect tool for backups:

- Differential copy: This means it will only copy the bits that have actually changed.
- On-the-fly compression: This type of compression makes the backups fast and consumes less bandwidth.

Advanced Tip:

If you want to install and update software on multiple systems on a network, you can save a lot of bandwidth and time by creating a local software repository.

On systems which use Zypp/YUM, you can create a local software repository using the following steps:

Firstly, mirror your desired repo to a folder, eg /var/www/ludsuserepo/rpms.

sudo zypper install createrepo

createrepo /var/ludsuserepo/rpms

At this point, all the required metadata will be added to the folder to make it a valid repository. To add this repository to the remote systems, you can use:

zypper addrepo -t YUM http://<host>/ludsuserepo/rpms local_repo

- Security: You can use the Secure Shell protocol (SSH) to do the backups, which makes the process of backing up very secure.

- Easy to use: rsync is very easy to use, almost like the cp command but with better features.

To do a local backup:

**# rsync -azvv <foldertobackup>
<destinationfolder>**

To do a remote backup over SSH (this will require OpenSSH server to be installed and started on the remote system):

**# rsync --delete -azvv -e -ssh /source/folder
user@remotemachine:/destination/folder**

Here's a breakdown of the options we've used:

- a preserves the timestamps and permissions of the files
- z compresses the data
- vv verbose output
- e sets the shell use for the transfer. Here we are specifying the SSH shell.

You can put these commands to the crontab file for regular differential backups.

System monitoring

Monitoring is an important part of system administration. It allows you to proactively react to issues in real-time. Monitoring also gives cues on how to improve the performance of the system. The following are some of the most important command-line tools used in monitoring various components of the system...

top: Top provides a real-time view of the running system. It can be considered as one of the most versatile system monitoring tools out there. It displays summary information, a list of threads or processes, types of system memory, process status, CPU usage etc.

uptime: Uptime displays the duration for which the system has been up. It also displays how many users are currently logged on, along with the system load averages for the past 1, 5 and 15 minutes.

```
$ uptime
12:18pm up 12:22, 4 users, load
average: 0.00, 0.01, 0.05
```

Advanced Tip:

Apart from monitoring the system, you may want to monitor how individual applications are doing. Strace will help you do just that.

Advanced Tip:

While traditionally distributions have been using the Linux init daemon to manage services, it has been replaced with modern alternatives. The most popular alternatives are systemd and upstart. Systemd is the default on Fedora/Red Hat, openSUSE, Arch Linux etc. Upstart is the default on Ubuntu, ChromeOS etc. Both of these tools provide almost the same kind of benefits, like parallel service startup and on-demand service initialisation. Both systemd and upstart are backward compatible with the init system, so init commands will work just fine.

Systemd uses the systemctl command to manage services, whereas upstart usage the initctl command for that purpose.

```
For example, to start a service:
# systemctl start foo.service
# initctl start foo.service
```

```
dhcpc3:/etc/init.d # top
top - 12:06:04 up 12:09, 4 users, load average: 0.00, 0.01, 0.05
Tasks: 126 total, 1 running, 125 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 3102660 total, 931868 used, 2170792 free, 60328 buffers
KiB Swap: 2103292 total, 0 used, 2103292 free, 529376 cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 7884 kunal    20   0 130m  25m  17m  S   1.0   0.8   0:10.46  konsole
 3383 root     20   0 27656 4132 3380  S   0.3   0.1   0:30.11  vmtoolsd
    1 root     20   0 5896  3592 2152  S   0.0   0.1   0:01.15  systemd
    2 root     20   0    0    0    0  S   0.0   0.0   0:00.00  kthreadd
    3 root     20   0    0    0    0  S   0.0   0.0   0:00.59  ksoftirqd/0
    5 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  kworker/0:OH
    7 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  kworker/u:OH
    8 root     rt    0    0    0    0  S   0.0   0.0   0:00.00  migration/0
    9 root     20   0    0    0    0  S   0.0   0.0   0:00.00  rcu_bh
   10 root     20   0    0    0    0  S   0.0   0.0   0:00.82  rcu_sched
   11 root     rt    0    0    0    0  S   0.0   0.0   0:00.19  watchdog/0
   12 root     rt    0    0    0    0  S   0.0   0.0   0:00.16  watchdog/1
   13 root     20   0    0    0    0  S   0.0   0.0   0:00.55  ksoftirqd/1
   14 root     rt    0    0    0    0  S   0.0   0.0   0:00.52  migration/1
   16 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  kworker/1:OH
   17 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  cpuset
   18 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  khelper
   19 root     20   0    0    0    0  S   0.0   0.0   0:00.00  kdevtmpfs
   20 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  netns
   21 root     20   0    0    0    0  S   0.0   0.0   0:00.00  bdi-default
   22 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  kintegrityd
   23 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  kblockd
   24 root     0 -20   0    0    0  S   0.0   0.0   0:00.00  md
   27 root     20   0    0    0    0  S   0.0   0.0   0:00.00  khungtaskd
   28 root     20   0    0    0    0  S   0.0   0.0   0:00.00  kswapd0
```

Output from the top command

sysstat performance tools: Most distributions do not include sysstat by default, but you can easily install it using your distribution's package manager. Sysstat includes the following tools:

- iostat:** reports CPU utilisation and disk I/O statistics;
- mpstat:** reports global and per-processor statistics;
- pidstat:** reports statistics for Linux tasks (processes);
- nfsiostat:** reports I/O statistics for network file systems;
- cifsioat:** reports I/O statistics for CIFS file systems

sar: collects and reports system activity information; These tools are very helpful in monitoring I/O across the whole system.

\$ iostat

```
avg-cpu:
%user %nice %system %iowait %steal %idle
0.85  0.05  2.51  6.14  0.00  90.44
```

Device:

	tps	kB_ read/s	kB_ wrtn/s	kB_ read	kB_ wrtn
sda	35.79	613.38	38.54	519671	32648
fd0	0.00	0.01	0.00	8	0

mpmap: mpmap reports a memory map of

\$ sudo mpstat

12:47:37 PM CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%idle
12:47:37 PM all	0.82	0.05	2.29	5.81	0.00	0.10	0.00	0.00	90.92

\$ pidstat

12:48:41 PM	PID	%usr	%system	%guest	%CPU	CPU	Command
12:48:41 PM	1	0.01	0.17	0.00	0.18	0	init
12:48:41 PM	2	0.00	0.00	0.00	0.00	0	kthreadd

\$ pmap -d 3275

Address	Kbytes	Mode	Offset	Device	Mapping
0000000000400000	900	r-x--	0000000000000000	008:00001	bash
00000000006e0000	4	r----	00000000000e0000	008:00001	bash
000000000025fc000	2076	rw---	0000000000000000	000:00000	[anon]
00007f0e5f20b000	2044	-----	000000000000c000	008:00001	libnss_files-2.15.so
mapped: 26960K writeable/private: 2356K shared: 28K					

Advanced Tip:

If you are looking for a more advanced solution for backup you can use Bacula (www.bacula.org). It is a fully fledged open source network backup solution. It also has its own ecosystem of add-ons which includes everything from specialised monitors and report builders to even a Bacula-specific file system (BaculaFS).

a process. It is very helpful in detecting memory bottlenecks.

```
$ pmap -d 3275
```

iptraf: iptraf is a TCP/UDP network monitoring utility. It has a nice ncurses-based user interface which liberates users from having to remember any command-line switches.

strace: strace intercepts and records the system calls which are called by a process and the signals which are received by a process. The name of each system call, its arguments and its return value are printed on standard error or to the file specified with the -o option. Strace is a useful diagnostic, instructional and debugging tool. It is particularly good for solving problems with programs for which the source is not readily available, since they do not need to be recompiled in order to trace them.

```
$ strace wget www.rarlab.com/rar/winrar-x64-420.exe
execve("/usr/bin/wget", ["wget", "http://www.rarlab.com/rar/winrar..."], [/* 43 vars */]) = 0
brk(0) = 0x2463000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
munmap(0x7f259cb5f000, 4096) = 0
stat("/home/kunal/.wgetrc", 0x7fff01fb9010) = -1 ENOENT (No such file or directory)
write(2, "Connecting to www.rarlab.com (ww"... , 67Connecting to www.rarlab.com (www.rarlab.com)|188.138.1.135]:80... ) = 67
socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(80), sin_addr=inet_addr("188.138.1.135")}, 16) = 0
write(2, "connected.\n", 11connected.
```

As you can see in the above example, we are using strace to obtain detailed information about everything wget is doing since we have issued the command. This includes the files it has opened, network connections it has made and so on.

Distribution-specific GUI administration/monitoring tools

While command-line and web-based administration are very powerful, GUI administration tools are easier and simpler to use. In this section we will look at some of the best GUI administration tool available on modern Linux distributions.

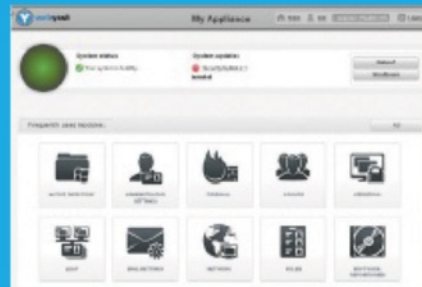
YaST2:

YaST (Yet another Setup Tool) is the installation and configuration tool for SUSE Linux distributions. YaST was one of the first to introduce a centralised configuration tool rather than having many single application utilities. YaST is an all-in-one solution which allows users to configure every aspect of a system, including managing packages, printers, sound system, kernel, partitioning, users etc. Configuration options are categorised under Software, Hardware, System, Network Devices, Network Services, Security and Users, Support, and Miscellaneous. All the configuration utilities provide an easy-to-use wizard-based interface. All YaST2 modules contain a dynamic help button for users who want more information on the configuration they are performing.

One of the key features that set YaST apart is its curses-based easy-to-use interface. It is very helpful for people who want to use all the power of YaST in text mode. YaST also includes a Ruby-based web interface called WebYaST, which provides all the features of YaST over the web.

YaST2 uses a modular architecture and additional modules can be developed using the YaST2 SDK.

YaST2 is included in all openSUSE Linux distros (as well as the commercial SUSE ones).

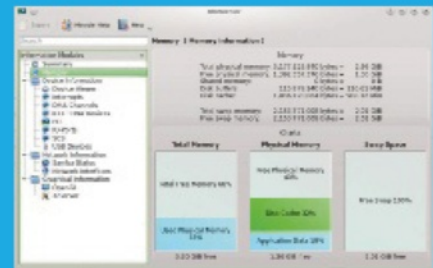


■ YaST2 curses-based text interface

KInfoCenter

KInfoCenter is KDE utility which provides hardware and graphics information. Most of this information is directly polled from the Linux kernel's /proc file system.

KInfoCenter is included in the standard KDE Software Compilation.

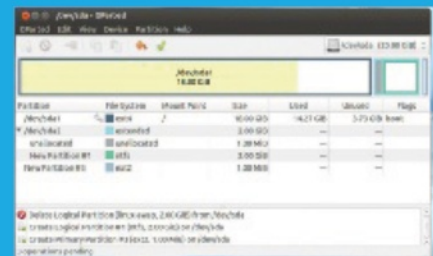


■ KInfoCenter

GParted

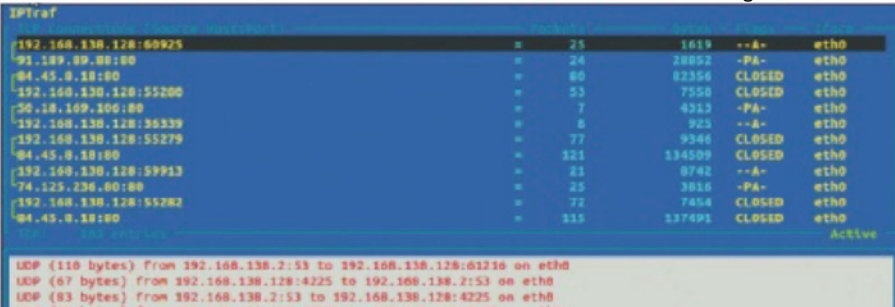
Parted is an excellent disk partitioning tool, but when it's not used carefully it can destroy data. That's where GParted comes in. It is an excellent GUI interface to Parted. It is easy to use and looks almost like the insanely popular Windows software Partition Magic. GParted uses a workflow-based approach to disk partitioning. Modifications are not applied automatically – instead, the user gets a chance to review the changes and can apply them only if he or she is comfortable with it.

GParted is included with the GNOME Software distribution.



■ GParted Partition Manager

■ IPTraf monitoring TCP Connections



Using a system administration configuration suite

In this section, instead of focusing on individual tools we will look at a solution which gives a full set of tools for system administration in one place.

System admin using Webmin

Webmin is a web-based system administration tool for a variety of UNIX-like systems. Webmin also has a vibrant ecosystem of modules around it. These modules extend the feature of Webmin to cover new applications and services.

Webmin is available for all the major Linux distributions. You can download it from: www.webmin.com/download.html

The easiest way to install it is from your distribution's package manager. If it is not

available in the package manager, you can download a DEB or RPM package from the Webmin site downloads page and install it directly on your system. After installing Webmin, it is available at <https://localhost:10000>. Here you'll need to log in with the root credentials. If you are using Ubuntu, then you will need to create a root password. You can create a root password using the following command:

```
$ sudo su
# passwd
```

1. Managing services

Expand System on the navigation bar, then click on Bootup and Shutdown. Here Webmin will list the type of boot system in use and all the services. It will also show if the service will start at boot and its current status.

Clicking on any service will open the service script. You can make changes to the service



■ Webmin default page

script and set its boot-time status. You can also start or stop the service from here.

2. Managing processes

Expand System, click on Running Processes. Here you'll see the Running processes list with process ID, Owner, Process Start Time and Command. Using the Display option, you can view user, memory and CPU usage as well.

Click on the process ID that you want to view/edit. On the process information screen, you can see the command, process information, owner and size among other details. You can use this screen to trace the process, see its open files and connections, or kill the process.

3. Configuring Apache web server

Traditionally, configuring Apache web server means editing the httpd.conf file. Webmin makes it very easy to configure Apache web server by providing a nice GUI interface to the Apache configuration files. To configure Apache web server, expand Servers on the navigation bar, then click on Apache Web Server. By default it will open in the Virtual Hosts. If you want to change Global Configuration, you can click on the Global Configuration tab. Click on a

Selecting the user interface for system administration

Command-line tools

Positive:

- ✓ They are easily accessible from within the system or remotely (using SSH or telnet).
- ✓ They can be also be used on a system with a low amount resources and are very handy in recovering a system which has only a command-line interface available.
- ✓ Command-line tools are easy to automate using scripts.

Negative:

- ✗ They are complex and more difficult to use than their web or GUI counterparts.

GUI tools

Positive:

- ✓ They are very easy to use and are often included with the distribution you are using.
- ✓ When designed properly, they give access to most options and provide automatic help and documentation right from the user interface.

Negative:

- ✗ Difficult to access from a remote system.
- ✗ Hard to automate.
- ✗ For each Linux distribution you may need to use different set of tools.

Web tools:

Positive:

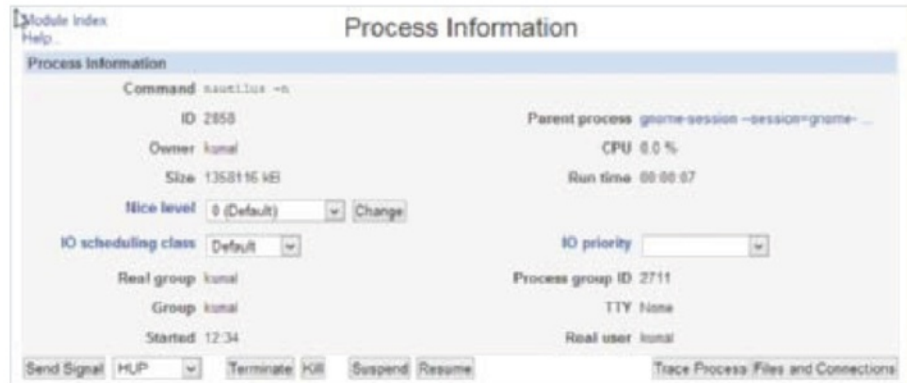
- ✓ Easy to use.
- ✓ Can be accessed remotely.

Negative:

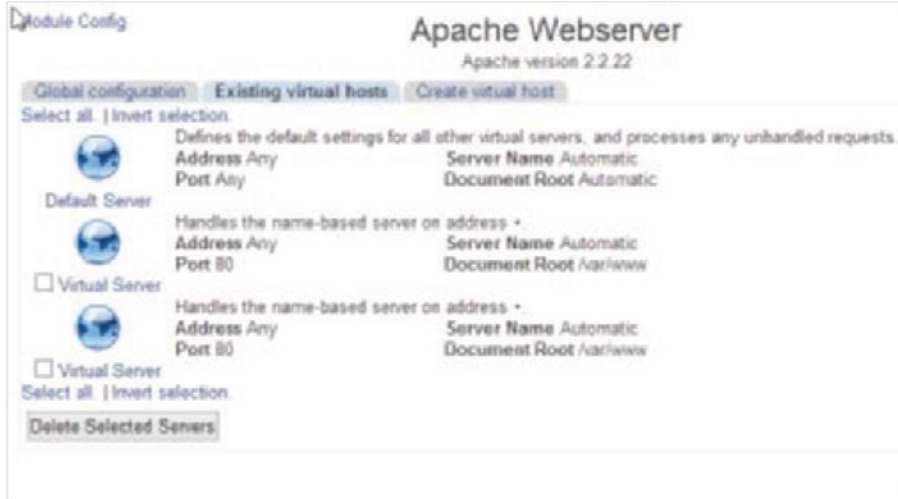
- ✗ Hard to set up.
- ✗ Security hole when not configured properly.

ID	Owner	Started	Command
1	root	12:32	lsbinit
455	root	12:33	sshd: root@kali: sshd -d
464	root	12:33	rsyncd --daemon
816	root	12:33	lsbinit --daemon
817	root	12:33	lsbinit --daemon
438	root	12:33	lsbinit --daemon
584	syslog	12:33	rsyslogd -c
639	messagebus	12:33	dbus-daemon -system -fork
721	root	12:33	sshd: root@kali: sshd -d
731	avahi	12:33	avahi-daemon: running [kali:localhost]

■ Running processes list



■ Detailed process information



■ Apache web server configuration

Virtual Host to modify it. Here you can configure options related to the virtual server, such as directory, MIME types, port, server name etc. Creation of a new virtual server configuration is also very easy: you can click on the Create Virtual Host tab to create a new Virtual Server Configuration.

4. Special features

Apart from system configuration features, Webmin also provides a few utilities which are excellent for new system administrators...

File Manager: Webmin comes with a built-in fully featured file manager. It is excellent for admins who want to make changes to the file system on the server. File Manager also comes with a handy editor which is excellent for making changes to configuration files. File Manager can be accessed via Others>File Manager. Note that File Manager requires a Java plug-in to be enabled on the browser side.

Built-in terminal: Most system admins would really appreciate having shell access to the server. But it is not always available everywhere. Webmin includes a nice little utility called Text Login which provides shell access to the server. It can be run on any browser and does not depend on Java. To access the shell, click Others>Text Login. Keep in mind that some systems do not allow root login from a remote shell. In this case you will need to use a regular user for login and then use su for performing administrative tasks.

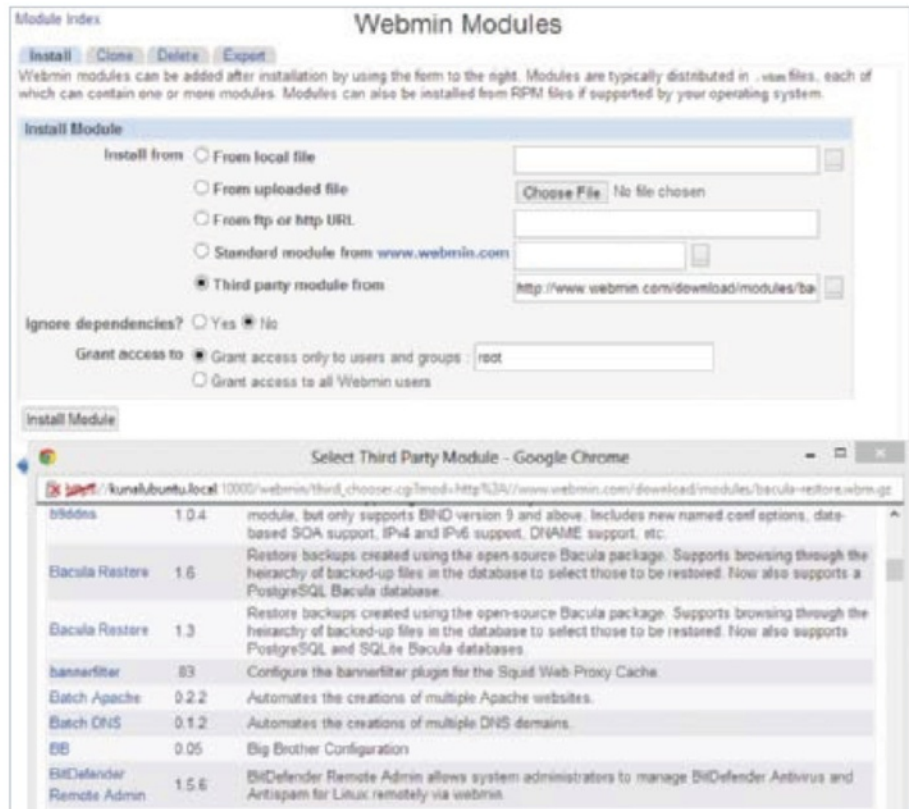
Webmin modules: Webmin has a thriving community of module makers. You can use these modules to add features to Webmin. Installing Webmin modules is very easy. Go to

Webmin on the navigation bar, then click Webmin Configuration>Webmin Modules. Here you can install both standard Webmin modules and third-party ones. Both options provide an automatic listing of modules. Just click on '.' and then on the module you want to install, and click Install.



Conclusion

There are some pretty fat books written about Linux system administration. This article was not an attempt to create an all-in-one guide, but a humble attempt to cover important things and get you excited to explore more.

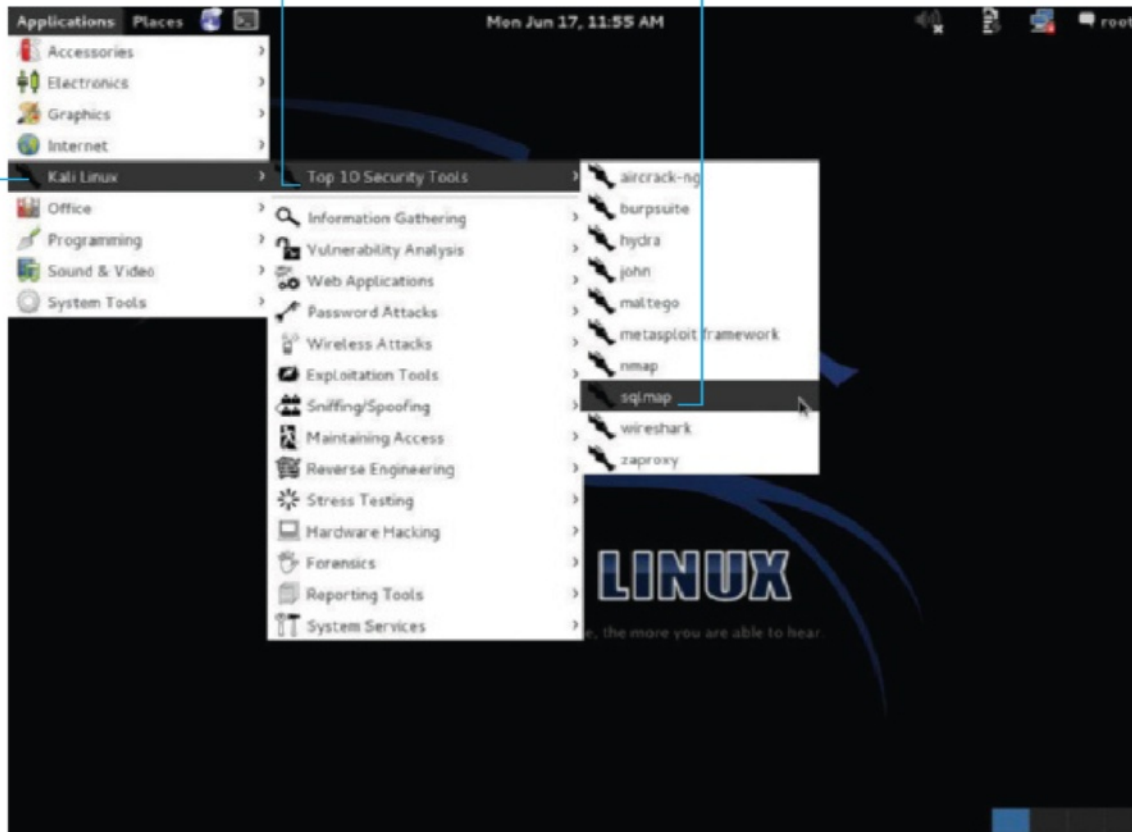


■ Webmin modules

Kali starts up with a top-level menu entry. Almost all of the tools available will be listed here, making it easy to start testing your security

The top ten applications that are used most often have their own menu entry. This saves you having to hunt in the submenus

Each application has an entry in the menu. If it is a console-based application, it opens in a new terminal with a listing of the options for that tool



Test your network's security

One of the best ways to test your security is to try to tear it apart, and you can do just that with Kali Linux...

Security is something that everyone needs to be aware of and something that everyone needs to deal with. While you can go out and collect a number of tools and utilities to help you out, there is an easier path. There are several Linux distributions out there that provide an entire suite of tools to fit your security needs. One of the more popular ones is Kali Linux (originally BackTrack). There are other ones, like BackBox or Lightweight Portable Security, which may fit specific needs better. You can run these off of a bootable DVD

or USB drive, allowing you to run forensics on a compromised machine. Alternatively, you could install it on a box and set it up on your network for a more permanent security solution.

In this tutorial, we'll use Kali Linux to go through one possible set of steps to analyse and test your local security. We will only be able to cover a subset of all of the tools available in Kali Linux, but you will learn some basic techniques to monitor your systems and to test your defences of them.

Resources

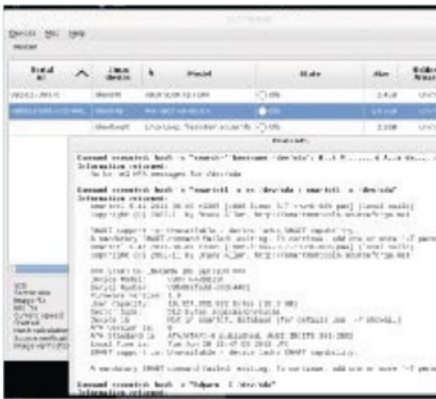
Kali Linux: www.kali.org

Metasploit: www.metasploit.com



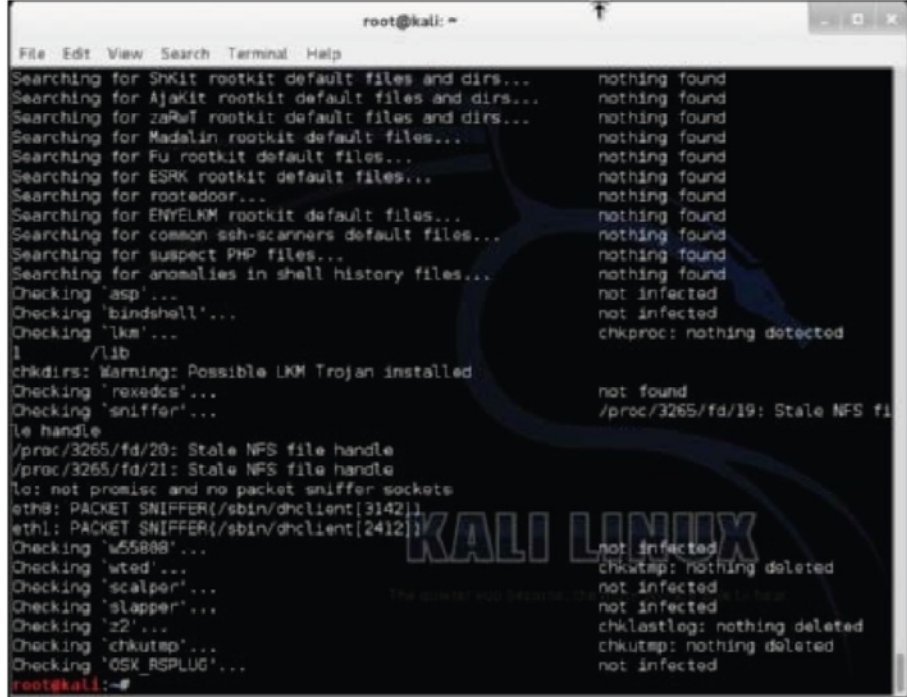
09 Forensics mode
If you do find a machine that you think may have been compromised, you want to be careful when you try to investigate it. Kali Linux provides a forensics mode on bootup that simply boots up and leaves all local drives unmounted and untouched. That way, you can run tests without changing the state of the system.

10 Offline password cracking
One of the things you will want to investigate is if the machine has been compromised due to weak password selections. There are several tools that can be used to try to crack password hashes. Most of these, like John the Ripper, use dictionary attacks to dig out passwords.



11 Guymager
In some cases, the machine in question may be too important to leave offline. In these cases, the only option is to make an image of the drive to investigate later before rebuilding. Guymager is one of the tools available to make images for this purpose.

“Once you have your network secured, that is only the beginning”

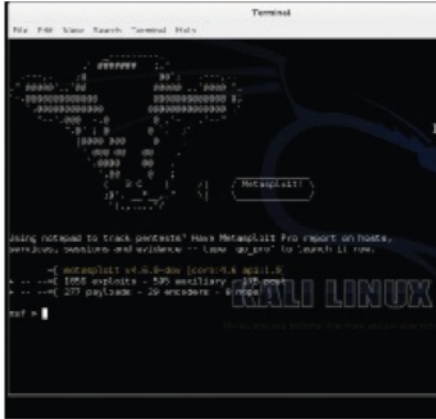


12 Chkrootkit
One of the things you will need to look for during an investigation is whether a rootkit has been installed, providing a back entrance to the bad guys. One of the tools you can use to do this is chkrootkit. This utility looks for evidence of common rootkits used for taking over machines.

13 Social engineering
One aspect of security that gets neglected is the social aspect. All of the security in the world won't help if your users aren't computing safely. Kali Linux provides a social engineering toolkit that you can use to do things like trying out spear-phishing attacks.



14 Exploit databases
Along with testing the social aspect, you also need to test the security of the machines to find any holes. You do this by trying known exploits. Luckily, there is a database full of known exploits online.



15 Metasploit

The usual tool used to test a system is Metasploit, which provides a full framework for putting together complete attack vectors. These include intrusions, compromises and channels to allow for remote access of a compromised machine. Within Kali Linux, there are menu items that allow you to start up the Metasploit server. There's also an entry to grab a dump of diagnostic logs, in case you run into issues. Metasploit runs in a client-server model, so once you start up the server, you will need to connect with a client in order to try some exploits against the machines that you are responsible for.



tool passively monitors a network to see what machines exist and what OS they run, without letting them know that you are listening.

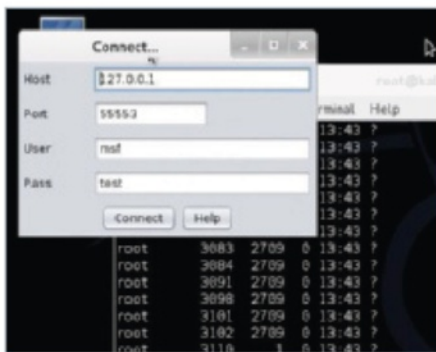


20 Android exploits – apktool

Once you have your Android device attached, you can run various exploits to get root access. These vary, based on what kind of hardware your Android is running on. One type of exploit may need apktool, in order to open and edit the APK files on your Android device.

21 Bluetooth

You also have another possible security hole. The Bluetooth protocol is used for mice, keyboards and other bits of hardware. But security was never really thought of in any major sense. Kali Linux provides several tools to look at the Bluetooth signals travelling around.



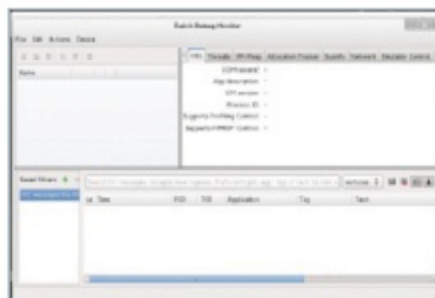
16 Armitage

One of the graphical interfaces available to you is Armitage. If you have already started Metasploit, then you can tell Armitage to connect to this already-running server. Otherwise, Armitage can start up a new Metasploit server for you to play with.



18 Hardware exploits

One set of tools that Kali provides that is unique is the ability to test other hardware. There are tools to poke into Android devices, Bluetooth protocols and Arduino systems.



19 DDMS

DDMS is a debugging monitor that gives you low-level access and control of Android machines. You simply need to plug your device into a USB port, start up DDMS and check out what is happening on the device. You do need to install an SDK for a specific version before starting.



22 Install on ARM

Support from the Kali developers has provided for an ARM architecture version. You can find it on the main download page. There are even instructions on how to install it on a Galaxy Note 10.1 device, including an installation image.

23 Conclusion

Hopefully, if you follow these steps, you can start to get a handle on the security needs for your system. This is only a start, though. There are lots more tools available in Kali Linux than we covered here, so don't be afraid to check out what else is available.

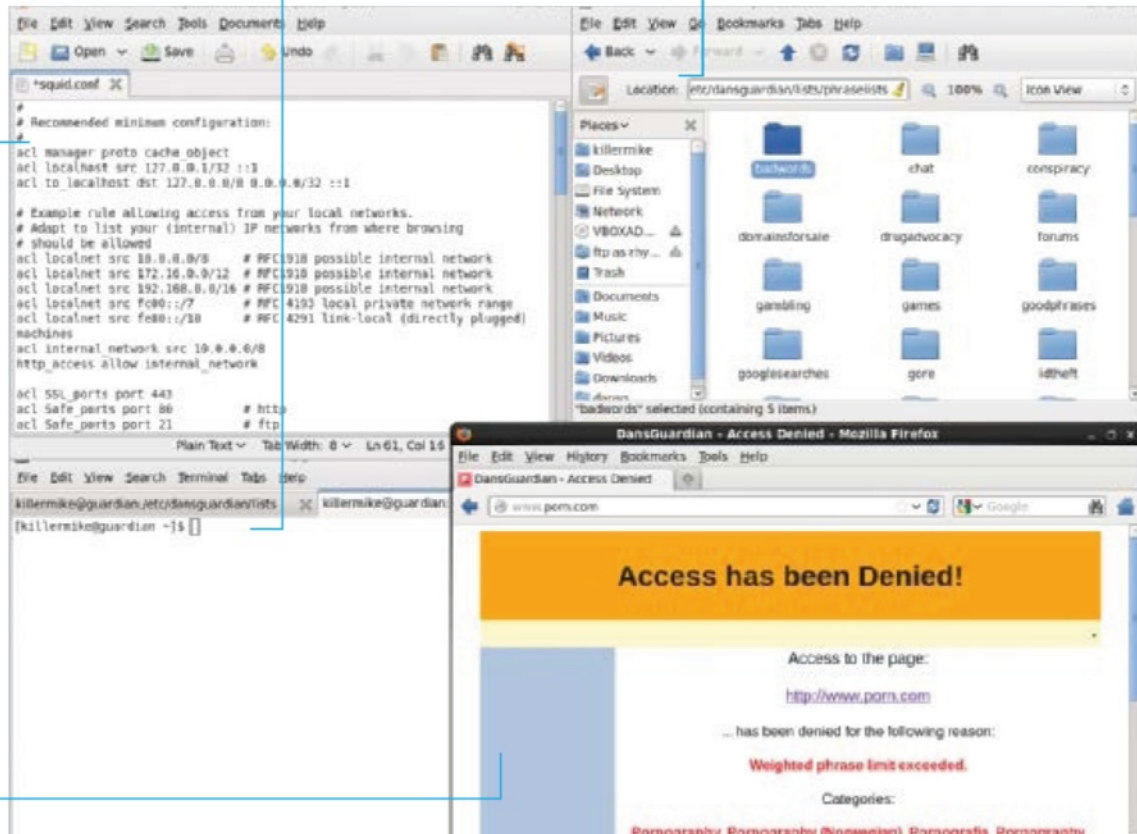
17 p0f

Once you have your network secured, that is only the beginning. You need to keep up with what is happening on your system. The p0f

Most of the configuration of the components is carried out using text files

We'll be spending a bit of time at the command line for this one

We're basing this project around a fresh installation of CentOS 6, but most of it can be applied to other distros



The finished result is a system that filters out the type of material that you tell it to, in an intelligent way

Protect your network

Build a gateway server that can intelligently filter content and block access to certain websites from certain PCs

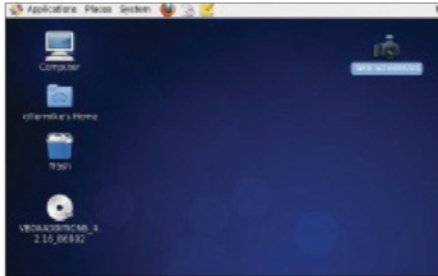
This is a project to create a gateway PC that allows you to filter internet traffic. We're going to use CentOS as the base of our system and the web filter DansGuardian will carry out the filtering for us.

Filtering the internet has never been more topical, and running DansGuardian puts that power into the hands of the administrator. Basic filtering software blocks individual pages, but DansGuardian is adaptive and analyses the content of pages on the fly. Even better,

DansGuardian carries out a sophisticated analysis of the content that uses weighted trigger phrases. This means that a single instance of a banned word might not block the page that the user is attempting to access.

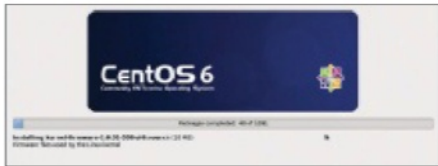
The gateway PC sits between your broadband internet connection and the rest of your network and is capable of assigning connection details to client PCs using DHCP. These computers will lack a direct connection to the internet until you configure them to use our proxy setup.

- Resources**
- Server machine
 - Two Ethernet adaptors
 - Firefox web browser



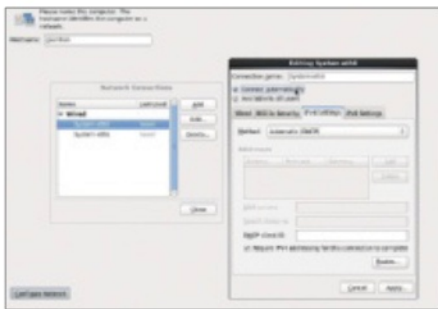
01 Set up server

Our example network layout revolves around a single server PC with two network adaptors – one connects to the internet (via router or modem) and the other to the rest of the network (via switch or hub). A Wi-Fi connection to outgoing connection is acceptable if it'll meet the bandwidth requirements of your network.



02 Install CentOS

Download the latest CentOS DVD image from www.centos.org. This installation is fairly standard until you get to the networking page. Give the computer a hostname, such as guardian, and then click on Configure Network.



03 Set up the adaptors

Click on a network adaptor, then on Edit... to edit the settings for each one in turn. Select the first adaptor and check 'Connect automatically'. Now select Method: Manual in the IPv4 tab. Give the first adaptor an address of 10.0.2.100, a netmask of 255.255.255.0 and a gateway corresponding to the IP address of your router. Give the second adaptor an IP address of 10.0.3.100. Accept the changes, then select Desktop installation profile and wait for the installation to complete. Upon reboot, create a basic user when prompted and then log in.



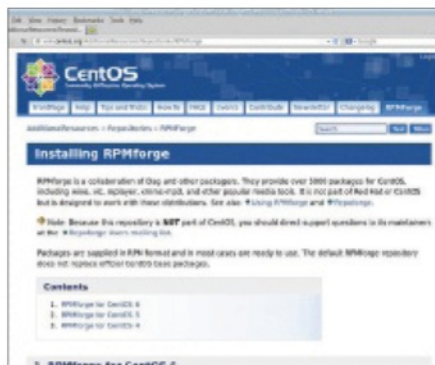
04 Become root

For most of this tutorial, you'll need to run as root. In CentOS, you can become root by typing `su` and then inputting the root password. For the bits that don't need root access, consider hitting Ctrl+T in the terminal window to create a tab with normal user access.



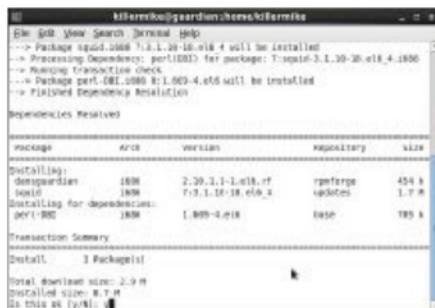
07 Start DansGuardian and Squid

We're going to use the service command to control all services. Start DansGuardian with `service dansguardian start` and then start Squid with `service squid start`. Check the output of both commands for errors.



05 Install the repository

Visit the CentOS RPMForge page (Google for it or go to tinyurl.com/4gjcxz) and follow the instructions there to download the `rpmforge-release` package. Install DAG's GPG key as instructed. Now install the package with `rpm -i [name of package].rpm`. Carry out a `yum update` to update the system.

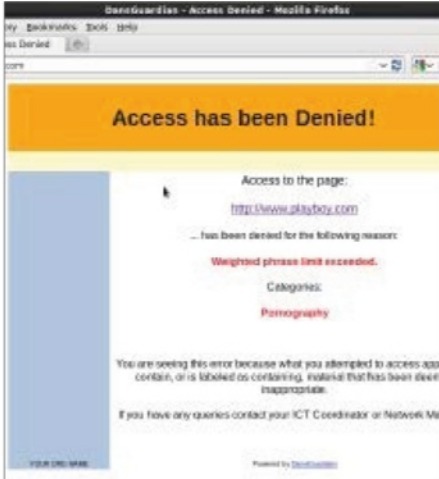


06 Install DansGuardian and Squid

DansGuardian and web cache Squid work in tandem with each other. Install them both by issuing the command `yum install dansguardian squid`.

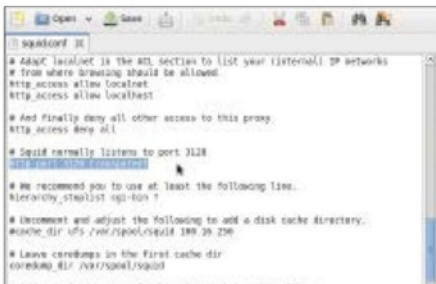
08 Test the proxy

Odds are, Squid and DansGuardian are working acceptably well with the default settings. To test this, we're going to select DansGuardian as the default proxy. Launch Firefox and go to Edit>Preferences>Advanced>Network. Now select the Settings... button. In the Connection Settings dialog, select 'Manual proxy configuration'. In the HTTP Proxy box, insert 127.0.0.1 with a port of 8080.



09 Test the proxy
Accept the changes you have just made and type `wikipedia.com` into the URL bar. If everything's working, the page should display as normal. If you're in a public place, choose a fairly tame site that should be blocked for testing. You should now see DansGuardian's default block page.

10 Configure Squid
Type `sudo gedit /etc/squid/squid.conf` & to open the Squid configuration. Add the lines `acl internal_network src 10.0.0.0/8` and `http_access allow internal_network`. In other words, process requests from machines with IP addresses that begin 10.x.x.x, which is our LAN. Add the line `visible_hostname guardian`. Type `service squid restart` to restart Squid.



11 Add DHCPD
Type `yum install dnsmasq`. Machines connected to the eth1 subnet need to be

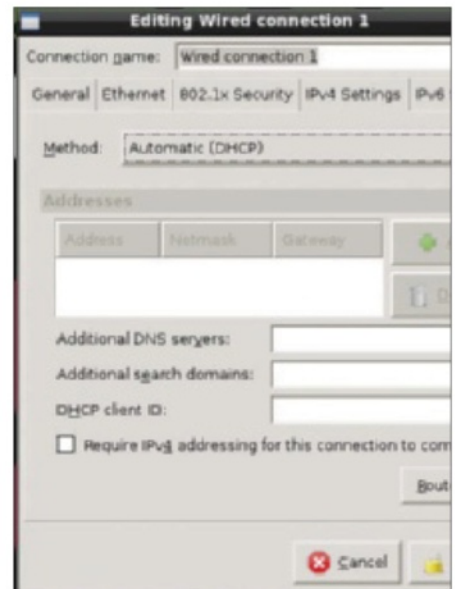


assigned an IP address. Edit `/etc/dnsmasq.conf`. Add the lines (without comments)...
`interface=eth1 #Only activate on the LAN`
`dhcp-option=eth1,3,10.0.2.100 #Specify the gateway`
`dhcp-range=eth,10.0.3.10,10.0.3.200,255.255.255.0,24h # Assign IP addresses 10.0.3.10 - 10.0.3.200.`



12 Configure services and restart
Type `chkconfig --add <service name>` followed by `chkconfig <service name> on`. Do this for the following services: `dnsmasq`, `dansguardian`, `squid`. Now restart the machine.

13 Configure the clients
Connect a machine to your LAN and make sure DHCP is selected on the client. The machines on the LAN should be assigned an IP address on startup – confirm by typing `ifconfig` into a terminal. In Firefox, set up the proxy as before, but add 10.0.3.100 as the IP address and check 'Use this proxy server for all protocols'.

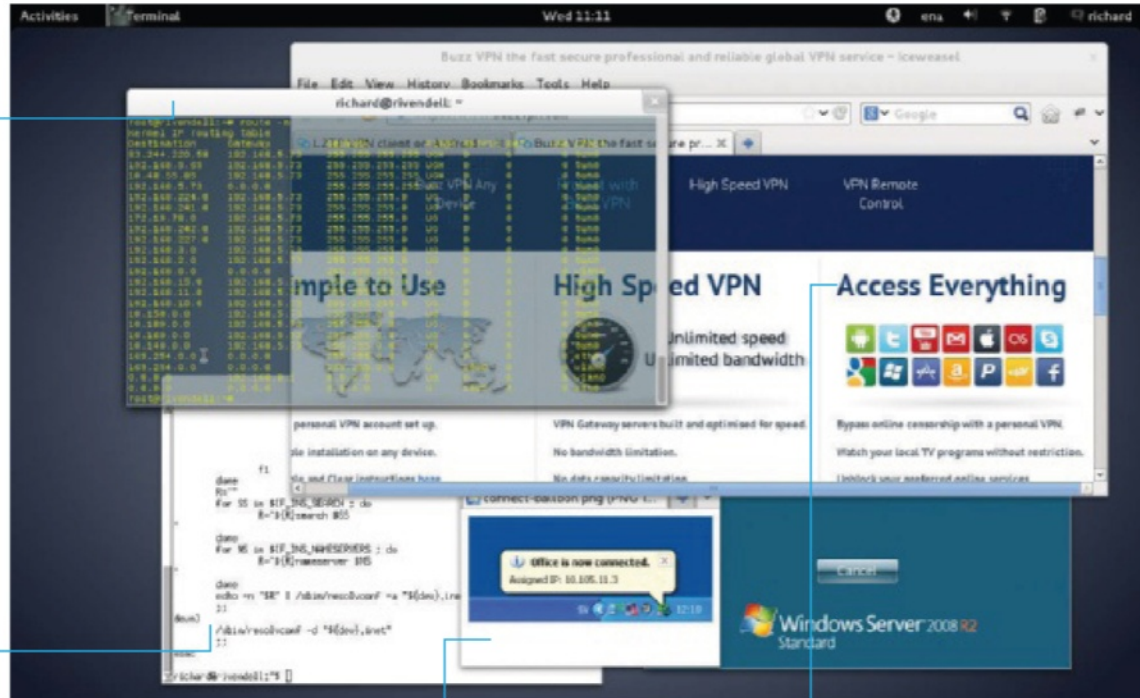


14 Configure DansGuardian behaviour
Most of the files that control the filtering behaviour of DansGuardian reside within `/etc/dansguardian/lists/` and you can guess many of

“Keep this list a secret and then assign a static IP to machines that require unfiltered access”

Connect to as many subnets and further VPNs as you would in your office

Push DNS to clients with a resolv.conf updater, as well as IP range and routing information, with simple server directives



Connect and disconnect easily from Windows clients, and admin via GUI

Protect your privacy and keep track of BBC iPlayer from across the globe

Configure a secure virtual private network

Stop worrying about SSH vulnerabilities and careless users – take control of who connects and how...

SSH offers astonishing flexibility to create ad hoc tunnels between networks, regardless of any firewall standing in the way. If this gets you re-evaluating the security of your network, and considering closing off SSH access from outside the network, in favour of restricted access to certain clients only then read on, as we show you how to configure a virtual private network (VPN) to allow only clients with pre-shared credentials to connect to your network.

VPN comes in many flavours, but here we will concentrate on OpenVPN (openvpn.net), which tunnels traffic via SSL and combines ease of setup with good functionality and presence across platforms.

While we are on the subject of planned remote connections, you will also want to take a look at VNC, to give users a full remote desktop experience rather than just a remote X Window. This graphical desktop sharing system enables running of software without font issues, for example, and easier access to Windows servers, as well as more complete access to the desktop for certain admin tasks.

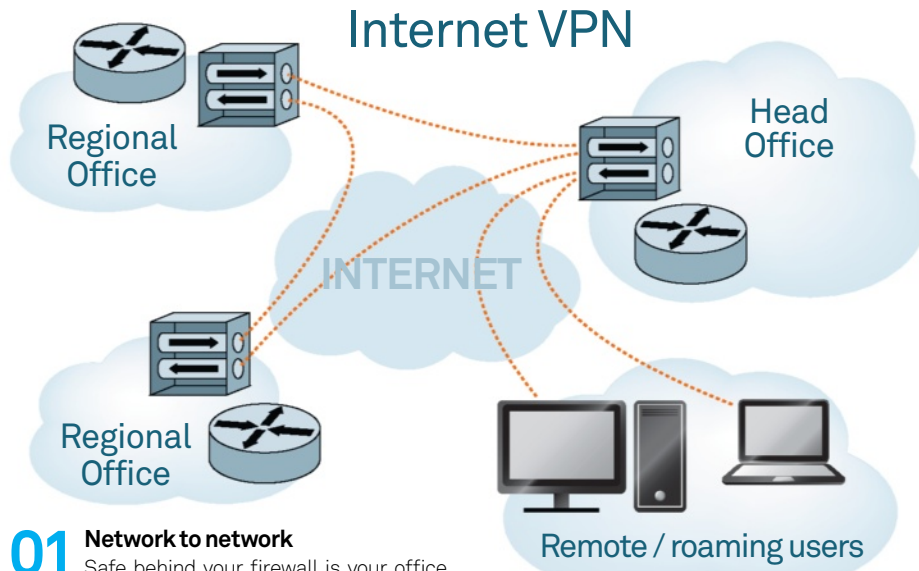
Rounding off, we must mention strongSWAN, which uses the IPSec extensions to encapsulate data securely at the datagram level (OpenVPN uses the good-enough-for-most-purposes OpenSSL – Secure Sockets Layer – library). Essential for the paranoid!

Resources

OpenVPN: openvpn.net

TightVNC www.tightvnc.com

strongSwan www.strongswan.org



01 Network to network
Safe behind your firewall is your office network; when you expand to another site, and another network, a VPN allows you to link the two (and further) networks as seamlessly as if they were plugged into the same router, and to give roaming users the same 'local' access.

```

root@kali:~# apt-cache show openvpn
Package: openvpn
Version: 2.2.6-8
Installed-Size: 1112
Maintainer: Alberto Garcia <alberto.garcia@netlab.es>
Architecture: i386
Source: openvpn (= 2.2.6-8)
Description: Virtual Private Network daemon
OpenVPN is an application to securely tunnel a protocol or a single UDP or TCP port. It can be used to access remote sites, make secure point-to-point connections, enhance wireless security, etc.
OpenVPN uses all of the encryption, authentication, and certification features provided by the OpenSSL library (any cipher, any size, or HMAC object).
OpenVPN may also utilize pre-shared keys or certificates (any exchange, if also supported by the selected encryption (ESP or Diffie-Hellman). Levels over 1024 or 2048-bit are supported. It will be such as Linux's ipsec,
Description-md5: 2e60304c3d8050e302d87b5d7
Tag: Network::VPN, Network::VPN, UserInterface::Network, Network::VPN, Network::VPN, Security::Encryption, User::Networking

```

02 OpenVPN
OpenVPN aims to be a universal VPN, and offers great flexibility, but is a relatively small download with few dependencies. It is able to work with passwords, certificates or pre-shared keys, using the OpenSSL library for its encryption capabilities.

```

root@kali:~# apt-get install openvpn real2key
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libpam0g
The following NEW packages will be installed:
  libpam0g
0 upgraded, 2 newly installed, 0 to remove and 33 not upgraded.
Need to get 636 kB of archives.
After this operation, 1,473 kB of additional disk space will be used.
Do you want to continue? [Y/n]

```

03 Easy install
Fire up a terminal emulator and `apt-get install openvpn` as root,

`sudo apt-get install openvpn` if you're on Ubuntu, or `yum install openvpn` for an RPM-based distro. Add OpenSSL if it's not already on your system, and resolvconf may be helpful.

04 Address: the problem
Before going further, let's consider one potential problem with routing: connecting from an internet cafe using the 192.168.0.0/24 subnet when your network uses the same. Something like 10.66.142.0/24 for your office network could save a lot of grief.

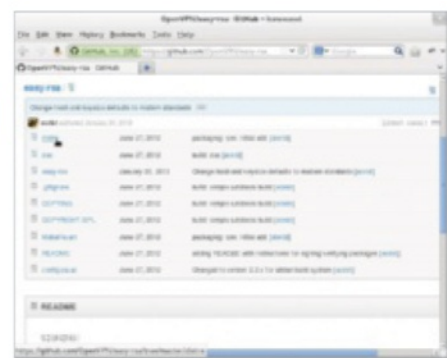
```

root@kali:~# ls /usr/share/doc/openvpn/examples/* -F --color
/usr/share/doc/openvpn/examples/
client.conf          openvpn-startup.sh*  tls-home.conf
firewall.sh*        openvpn-startup.sh*  tls-office.conf
howto.sgml          README               xirand-client-conf
loopback-client     server.conf.gz       xirand-server-conf
office.sgml         static-home.conf
/usr/share/doc/openvpn/examples/sample-config-files:
client.conf          openvpn-startup.sh*  tls-home.conf
firewall.sh*        openvpn-startup.sh*  tls-office.conf
howto.sgml          README               xirand-client-conf
loopback-client     server.conf.gz       xirand-server-conf
office.sgml         static-home.conf
/usr/share/doc/openvpn/examples/sample-keys:
ca.crt               client.crt            dh2048.pem            pass.key             server.key
ca.key               client.key            pkcs12.p12           server.crt           ta.key
/usr/share/doc/openvpn/examples/sample-scripts:
auto-pull.sh*        auto-pull.sh*        auto-pull.sh*
bridge-client*       openvpn-init.gz      verify.sh*

```

05 Simpler config
How do you keep a flexible app simple to configure? By including config examples to

modify. Grab the easy-rsa examples with `sudo cp -R /usr/share/doc/openvpn/examples/easy-rsa /etc/openvpn/`



06 Public-key infrastructure (PKI)
We're going to use easysrsa to create a master CA certificate, to sign the certificates which we'll generate for the server and each client. Recently easysrsa has been separated out from OpenVPN, so you may need to download it from github.com/OpenVPN/easy-rsa

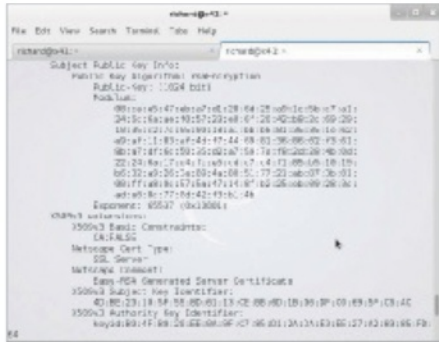
```

mc [root@x41]~/etc/openvpn/easy-rsa/2.0
# Edit View Search Terminal Tabs Help
chard@x41:~# mc [root@x41]~/etc/openvpn/easy-rsa/2.0
ls nano 2.2.6 File: /etc/openvpn/easy-rsa/2.0/var
sort CA_EXPIRE=3650
In how many days should certificates expire?
sort KEY_EXPIRE=3650
These are the default values for fields which will be placed in the certificate. Don't leave any of these fields blank.
sort KEY_COUNTRY="GB"
sort KEY_PROVINCE="Merseyside"
sort KEY_CITY="Liverpool"
sort KEY_ORG="World Domination"
sort KEY_EMAIL="bob"
sort KEY_EMAIL_MAIL="bob@host.domain"
sort KEY_CN=changeme
sort KEY_NAME=changeme
sort KEY_OU=changeme
sort PKCS11_MODULE_PATH=changeme
sort PKCS11_PIN=1234
Get Help WriteOut Read File Prev Page Out
Exit Justify Where Is Next Page Up Up

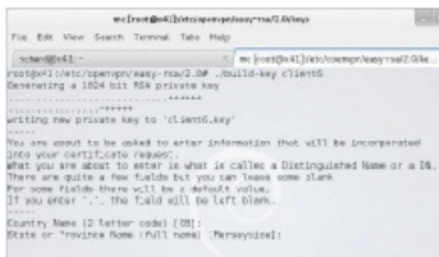
```

07 Master certificate
Edit the vars file, changing the KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG and KEY_EMAIL parameters. Other values that may need changing are usually helpfully marked as "=changeme" – both the comments and the README file will guide you.

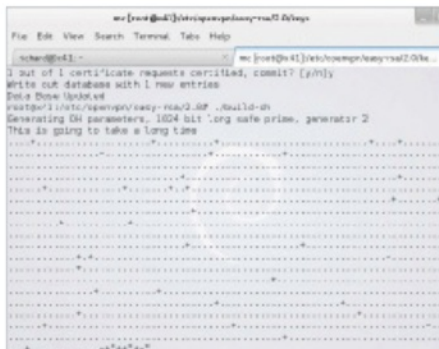
08 Generation game
From within the same directory as the vars file we have just generated – /etc/openvpn/easysrsa/2.0/ in this case – we run the build script. Note that instead of 'hostname' for Common Name, you may wish to enter OpenVPN-CA.



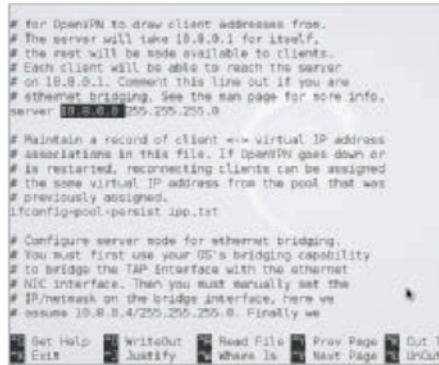
09 Build server certificate
Running `./build-key-server server` next differs slightly as 'server' is offered as the Common Name (accept this), then you are offered a challenge response (skip this), and to sign the certificate (choose yes).



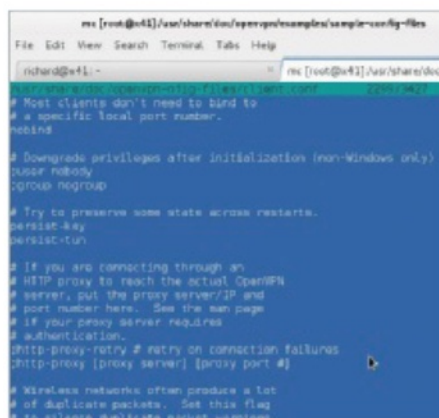
10 Roll out the client certs
Now build as many client certificates as you need with variations on `./build-key client1` – because each client certificate is signed with the same master certificate as the server key, the server will not need to keep copies of the client keys.



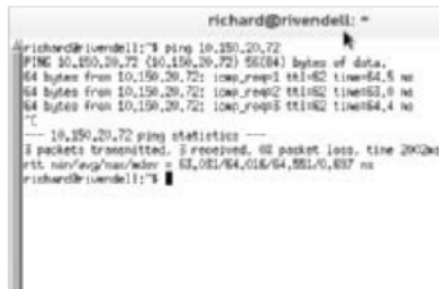
11 Diffie-Hellman
No, it's not a brand of mayonnaise! The Diffie-Hellman key exchange method "allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel." Run `./build-dh`



12 Server config
Start with the sample `server.conf` from the `/usr/share/doc/openvpn/` example configs. Change the address range from 10.8.0.0 to your own. Other options include the ability to push the route, eg: `push "route 10.13.101.1 255.0.0.0"`



13 Nearly there
On your client PCs, copy the keys you have generated (using scp or a USB key), and edit the sample `client.conf` file. Uncommenting the `user nobody` and `group nobody` directives will add to security. Now it's time to test...

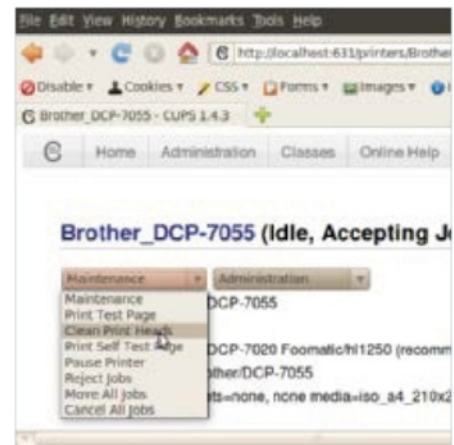


14 Is anyone there?
Start OpenVPN on the client with `openvpn path/to/conf`. From the client, try

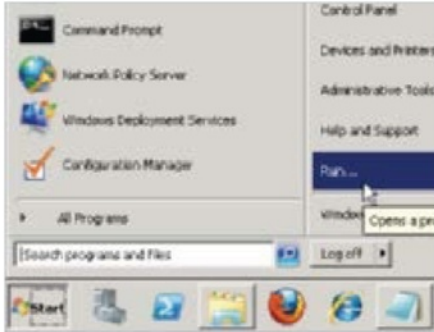
pinging an address on the remote network. Given correct address data, any errors are likely to be firewall-related. Success? Now start with `/etc/init.d/openvpn start`



15 We have tunnel!
`ifconfig tun0` (or `ifconfig tap0` if you're using a virtual Ethernet device instead of a point-to-point IP tunnel) will now show all the info, giving the addresses at each end of the tunnel. If you enabled the `push "route..."` and `push "route-gateway..."` directives in the server config, you will now be able to also reach whatever other networks are visible to the server via other VPNs, as shown in the opening screenshot of the article. The `push "dhcp-option DNS 10.66..."` directive may also be useful to you.



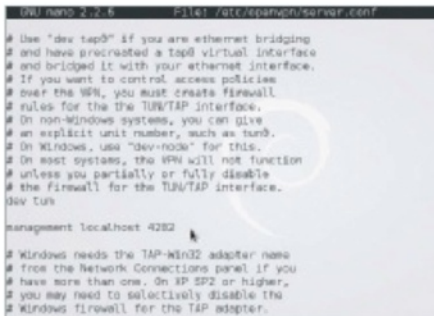
16 Remote access
Now you have your secure connection into the office, you'll want to do more than just ping boxes. You can roam the intranet, performing local admin tasks on printers and servers from the comfort of your favourite cafe...



17 Desktop sharing
Adding VNC into the mix will enable you to work with GUI apps on remote systems across the VPN, whether GNU/Linux, Windows or whatever. xvnc4viewer will give you more power than Ubuntu's built-in desktop, and TightVNC at both ends gets through narrow bandwidth connections.



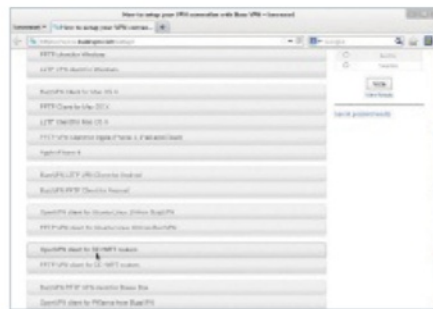
18 Spread the network
As well as clients for UNIX, Windows and even Maemo, there's an Android port of the client software at github.com/fries. Once upon a time OpenVPN was console-only admin on Windows, unless you went to openvpn.se; now it's all included in the package.



19 Admin tasks
Enabling management on the port of your choosing gives you access via **telnet localhost:4202** – from here you can disconnect clients; toggle logging; and perform tests and debugging. The management GUI accesses OpenVPN through this interface.



20 GUI choice
OpenVPN's popularity can be seen in the vast choice of third-party GUIs, both to OpenVPN itself (connection clients) and to the management interface. While proprietary bolt-ons are a familiar tale, FOSS options are available too.



21 Hassle-free VPN
If you just wanted a VPN to protect your browsing privacy, say, or to catch BBC iPlayer while overseas, then one of the many commercial VPN providers is a hassle-free alternative, with downloadable clients for nearly every device. Read the reviews to find a suitable one.



22 Security first
Alternatively, IPSec gives you secure encapsulation of your data inside an IPSec packet, aiming for authentication, integrity and confidentiality. It's favoured by government agencies, those fearing industrial espionage, and anyone else feeling justifiably paranoid.

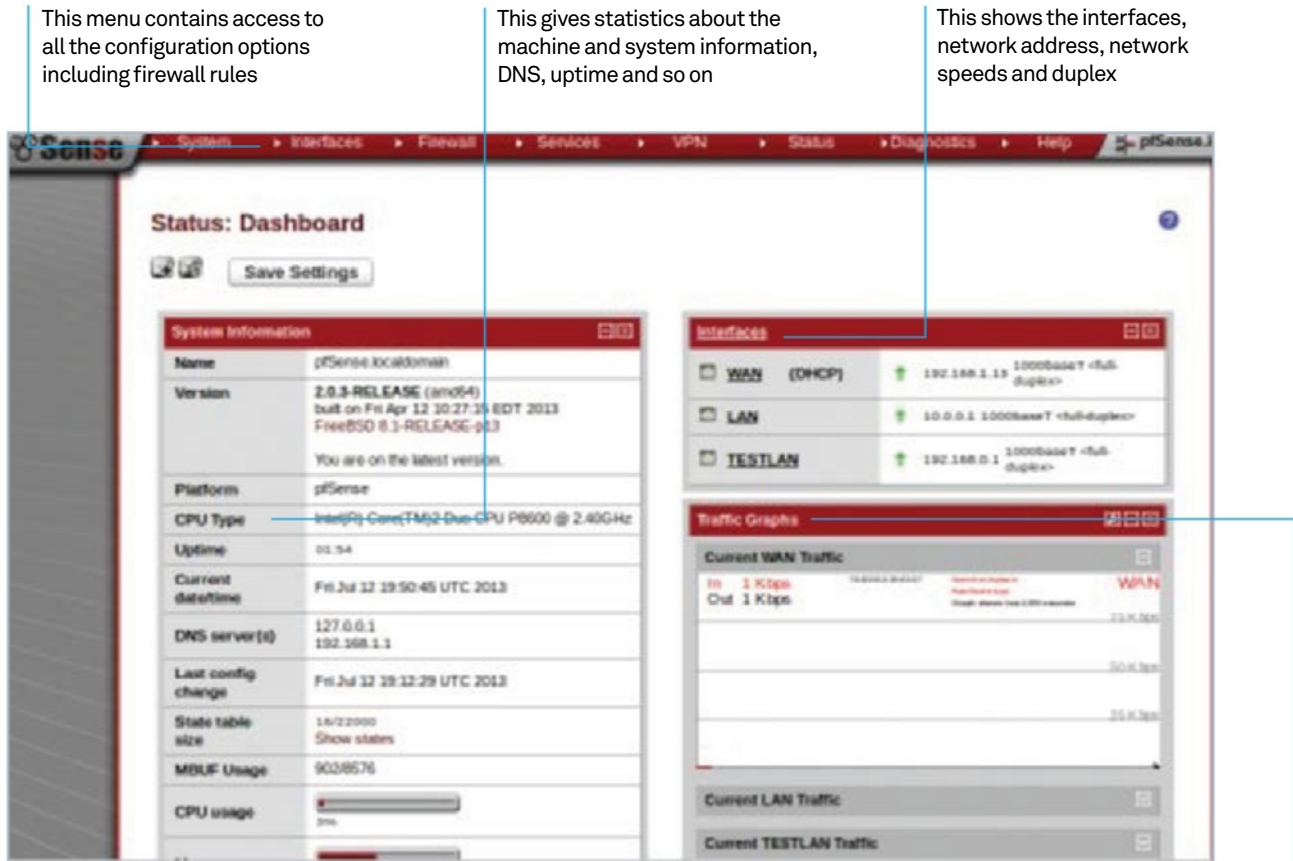
“Remote access with VPN saves opening networks to SSH tunnel’s firewall-defying antics”



23 Swanning through
For IPSec, strongSwan – a successor to FreeS/WAN (Free Secure Wide-Area Networking) – provides compatibility with other IPSec implementations, including clients on other platforms, combined with IKEv1 and IKEv2, and a good reputation for security.



24 Brain food
There's plenty of accumulated wisdom on remote access and admin. While the world wide web offers much that is useful, don't neglect print format! Some of the sysadmin manuals and server hacks books available contain some great tips for remote, secure admin and much more.



Build your own pro-grade firewall

The basic network I/O occurring through your firewall

Learn how to create a powerful multi-network hardware firewall with a redundant computer

Resources

A Linux PC with 3 network cards (min 300MHz, 128MB RAM)

pfSense live CD: www.pfsense.org

Labelling system for network cards

An ADSL or cable modem

A second Linux PC

This in-depth tutorial covers setting up a hardware-based firewall and configuring it to make it hacker resistant and business class.

It will cover the configuration of a basic two-network setup consisting of an internal network for all your test setups and a second LAN that can be used for normal everyday usage. We will include a DHCP setup on your second LAN to make your life that little bit easier.

The networks are to be configured in such a way that any breakages on your test network won't affect your normal network. This guide will

also cover creating a sensible rule base to which you can add extra rules if you wish. Additionally, you'll find tips and tricks to make everything more secure than a simple default setup. Finally, we will cover how to back up and restore your firewall configuration, should the worst happen.

If you want to just experiment with this without going the whole hog, you can do it within a virtual machine, two virtual networks and a bridged adaptor to your local network. The scope of this setup is outside the bounds of this article, but our walkthrough should still work perfectly.

```

Welcome to pfSense 2.0.3-RELEASE ...

Mounting unionfs directories...done.
Creating symlinks.....done.
Launching the init system... done.
Initializing..... done.
Starting device manager (devd)...done.

[ Press R to enter recovery mode or ]
[ press I to launch the installer ]

(R)ecovery mode can assist by rescuing config.xml
from a broken hard disk installation, etc.

(I)nstaller may be invoked now if you do
not wish to boot into the liveCD environment at this time.

(C)ontinues the LiveCD bootup without further pause.

Timeout before auto boot continues (seconds): 4
    
```

01 Install pfSense on your redundant PC

Boot from the pfSense live CD you downloaded and burnt in the prerequisites (see Resources). Allow it to boot up with defaults until you get to the screen that mentions recovery and installer. Press the I key to invoke the installer. Accept the defaults presented on screen by selecting 'Accept these defaults'. The only possible change you might want to make is to your keyboard layout if you have a non-US/UK-type keyboard. Now simply select Quick/Easy Install. Read the warning – the installation will totally destroy any information on the disk, so back up first if you want to preserve your data. When you're ready, select OK. Once the installation is done, select Standard Kernel and once that's configured, navigate to Reboot and press Enter. Make a note of the default username and password (admin/pfsense). Remove the CD and the machine should reboot into the network configuration menu where all the good stuff starts to happen.

02 Configure networking

At this point, make sure your network cables are not plugged in. After booting into pfSense you will see a basic text configuration screen and a list of the network cards installed. When asked if you wish to configure the VLANs, select no (by pressing N). Next we are going to auto-detect the network. To set up the WAN connection, press A. Now insert the WAN cable from your router into the first network port. You will see it change status to UP, then press Enter to continue. We have now configured the WAN port to the internet – repeat the same process for your first and second LAN cards in the same fashion. Once complete, press Enter to continue.

This finishes the installation and lets the firewall know there are no more network connections to be configured. Answer Yes

when asked 'Do you wish to proceed?'. It will now commit the settings to disk. It will also give you a list of networks to match up again your network cables. It is a good idea to label them up now to save confusion later.

03 Introducing the pfSense setup

After configuring the network connections and rebooting, you'll still see the CLI with a series of menu options. Since the other networks need to be configured and you can do this by pressing 2 on the console. You'll now see you can configure IP address setup for all the networks. Select the NIC that corresponds to your wireless or basic internal network. This is our (WIRELESS) LAN so let's give it 192.168.1.1 with 254 addresses. Enter the IP 192.168.1.1 –

this will become our gateway. This tutorial is using a /24 network, so type in 24 followed by Enter. It will ask if this network needs a DHCP server – select Yes. The configuration program will then ask about the start of the DHCP range. It's best to start at 192.168.1.2. Follow this with the end of the range, 192.168.1.32. This is up to you and depends on your needs, but 30 DHCP leases is more than enough. Press N on the HTTP protocol question. Repeat the process with the other network and select 10.0.0.1 as the interface address, 24 as the network mask and use the range 10.0.0.2 – 10.0.0.32.

04 Using the pfSense GUI

In this section we'll set up the basic GUI. Connect a laptop to the network of the WIRELESS LAN and open a web browser and enter <https://192.168.1.1> in your browser. You may receive a warning about an untrusted network connection, but that is fine to ignore for our purposes. This address and webpage is the network address (gateway) you configured earlier in the tutorial. It may be necessary to add an exception and hit Continue on your web GUI page.

You will be greeted with the setup wizard. Select Next to get started. At this point you can leave the hostname and network name alone, unless you want to put your own DNS servers in. If you leave the override DNS feature, you will get your DNS for your DHCP servers from your ISP.

```

      f
     / \
    /   \ Sense
   /     \
  /       \
 /         \

Welcome to pfSense 2.0.3-RELEASE ...

Mounting unionfs directories...done.
Creating symlinks.....done.
Launching the init system... done.
Initializing..... done.
Starting device manager (devd)...done.

[ Press R to enter recovery mode or ]
[ press I to launch the installer ]

(R)ecovery mode can assist by rescuing config.xml
from a broken hard disk installation, etc.

(I)nstaller may be invoked now if you do
not wish to boot into the liveCD environment at this time.

(C)ontinues the LiveCD bootup without further pause.

Timeout before auto boot continues (seconds): 4
    
```

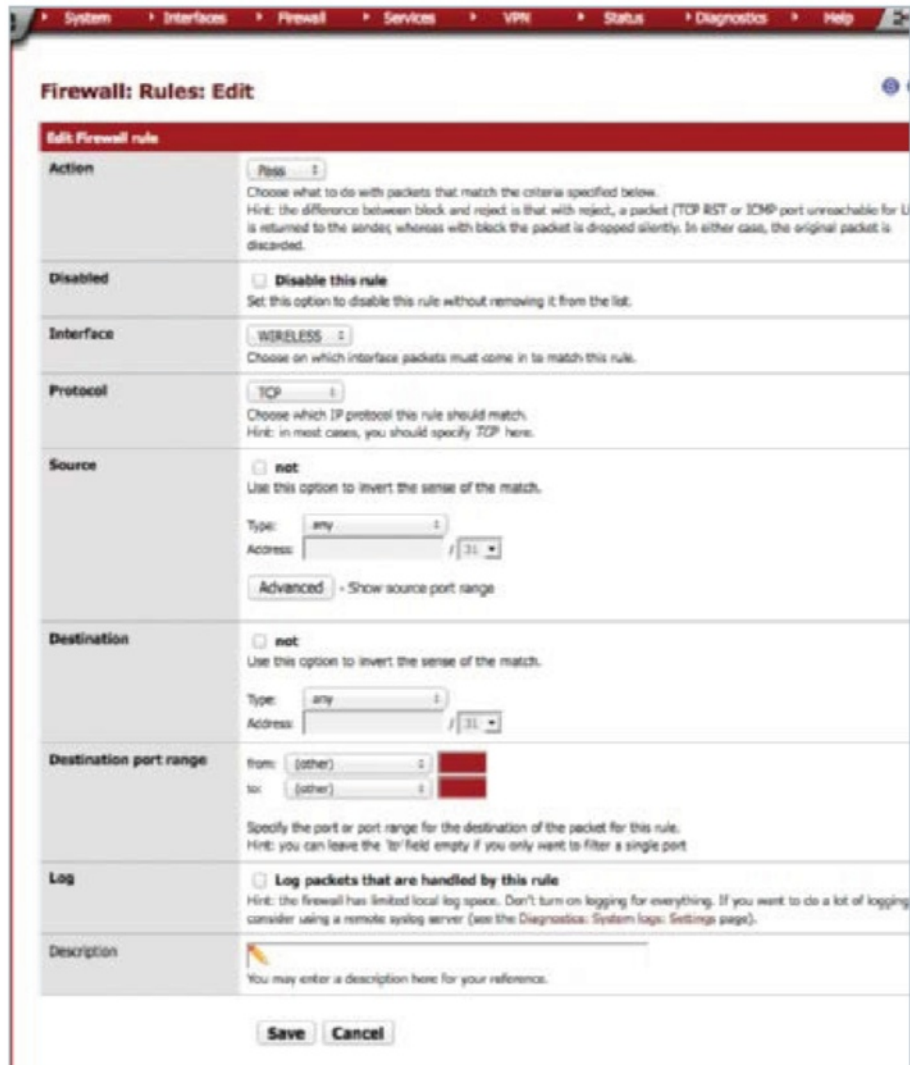
Configure the time servers and click Next. On the next page you can configure any extra setup information if your ISP requires it. Click Next to go to the LAN page. Lastly, change the admin password to a secure one of your choice. At this point the firewall will reload its rules. Enable the third network, click Interfaces>OPT1 and select 'enable interface' and click Save. Rename OPT1 to LAN by clicking on Interfaces>OPT1 and renaming it LAN.



05 How to create a basic rule All rules are added in the same way; just add and modify each rule to fit the requirements. Click the bottom left '+' symbol from the Firewall Rules page to start creating one. Now we can add web browsing. Set action to pass (unless you wish to set up a rule to drop traffic). Choose your source interface (LAN/WIRELESS). Follow this by selecting your protocol to use (usually TCP, but things like DNS require UDP port 53). On the next item, select the destination. Usually this will be the any address for external traffic and WIRELESS or LAN subnet for address, depending on requirements.

Destination port is straightforward enough: you can select a range of ports by either using the drop-down menus or entering your own ranges (for now, just select HTTP). Using multiple ports is covered later in the article.

One set of rules definitely needed for both networks is basic HTTP and HTTPS rules for browsing. You will also want to implement a 'drop all' rule. As the name implies, this drops all traffic. This makes sure no traffic escapes out of your network that you intended. To do this, just set up a rule that has drop for the action, networks and port ranges set to any TCP/UDP on the protocol. Do this for both networks.



06 Aliases make life easier Aliases enable you to group ports together. As the name suggests, they allow you to use an alias in your rules that can refer to groups of items. An example would be combining HTTP and HTTPS together in one alias. No need for multiple rules – just one alias can be used to ensure correct ports are opened!

From the Firewall menu, select Aliases. Use the '+' on the right. To implement HTTP and HTTPS together, give it a name like Web_browsing_ports – ensure it is descriptive. Select ports from the Type drop-down. Hit the

'+' button below the ports and add 80 in the port and HTTPS in description. To add HTTPS, click the '+' button, but use port 443. Save and apply changes. Aliases are not limited to ports, but can also be used for hosts and networks. To implement an alias in a rule (assuming the alias has been created beforehand) go to the Rules Port drop-down, select Other and begin to type the name of the alias. It should pop up a list. Just click on the alias needed and accept. Apply the changes once the rule is created. Similar rules can be created between networks. An example would be SSH. Implement this rule the same way

“No need for multiple rules – just one alias can be used to ensure correct ports are opened”



07 Enhanced rule sets

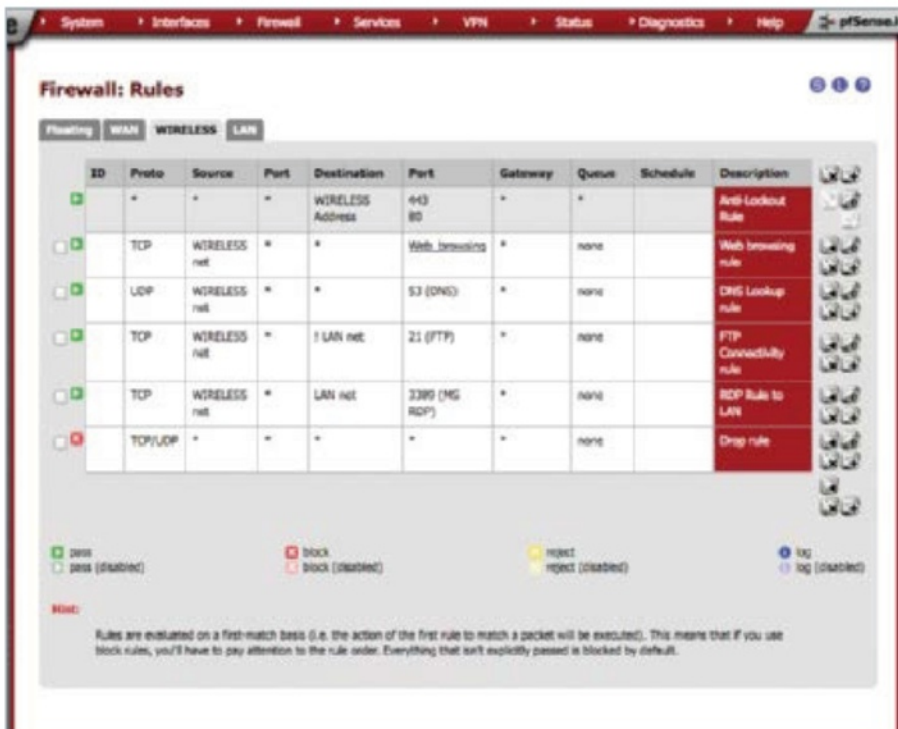
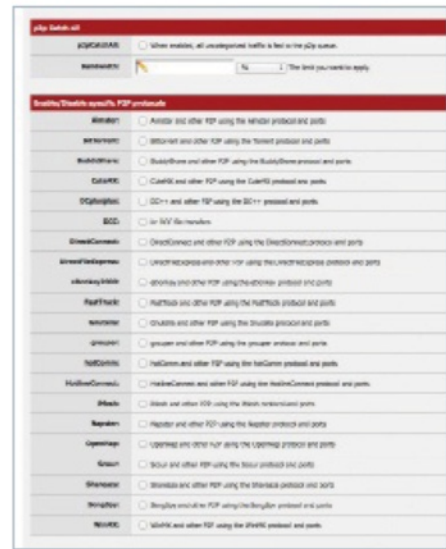
Now that you understand how basic rules work, it is time to group together a more enhanced rule set. As a minimum, set up both networks to have the following flowing out the internet. HTTP and HTTPS (remember to use an alias here!), include FTP, DNS (using UDP) as well as SSH if needed. However, box clever here. If you only use SSH to talk to a specific number of hosts, use an alias with the Hosts drop-down and enter the IP addresses into the alias. That way, should a machine be compromised, it will

not be able to talk SSH on port 22 to anything but those boxes defined in the alias. The more specific the rules, the more secure they are. You will also need to repeat the process on the LAN, assuming you want the same rights. To prevent a network talking to another on a certain port and protocol, use the NOT option in the rule base. An example would be to change the web browser rule to say destination NOT LAN – you will then find you can no longer browse any web server on the test network, but can browse the internet.

08 Managing the bandwidth

Now we can look at some other features such as bandwidth management. PfSense makes it easy to block file-sharing platforms such as BitTorrent, WinMX and similar. It can also split the bandwidth between the two networks. Do this by going to Firewall>Traffic Shaper. Click the Wizards tab. There are a number of different scenarios; select the 'Single WAN, Multi LAN' option. Enter number of LANs (two in this case) and press Next. Fill in your available download and upload speeds. Leave the other components and click Next. Unless you use SIP, click Next. Penalty box can be used to restrict specific groups or alias groups of machines to a percentage of the capacity if needed. Click Next. Use this page to lower the priority or even block P2P traffic completely.

Click Enable on the Traffic Shaper wizard and then select any protocols to allow/block. Edit to the preferred setup and then click Next. On this page, configure traffic shaping for games, with preconfigured optimal setups if needed. Finally you can do the same for applications if you wish to, such as RDP, VNC etc. Click Finish. To remove the shaping, go back to the Firewall Traffic Shaper menu and select 'Remove shaper'.

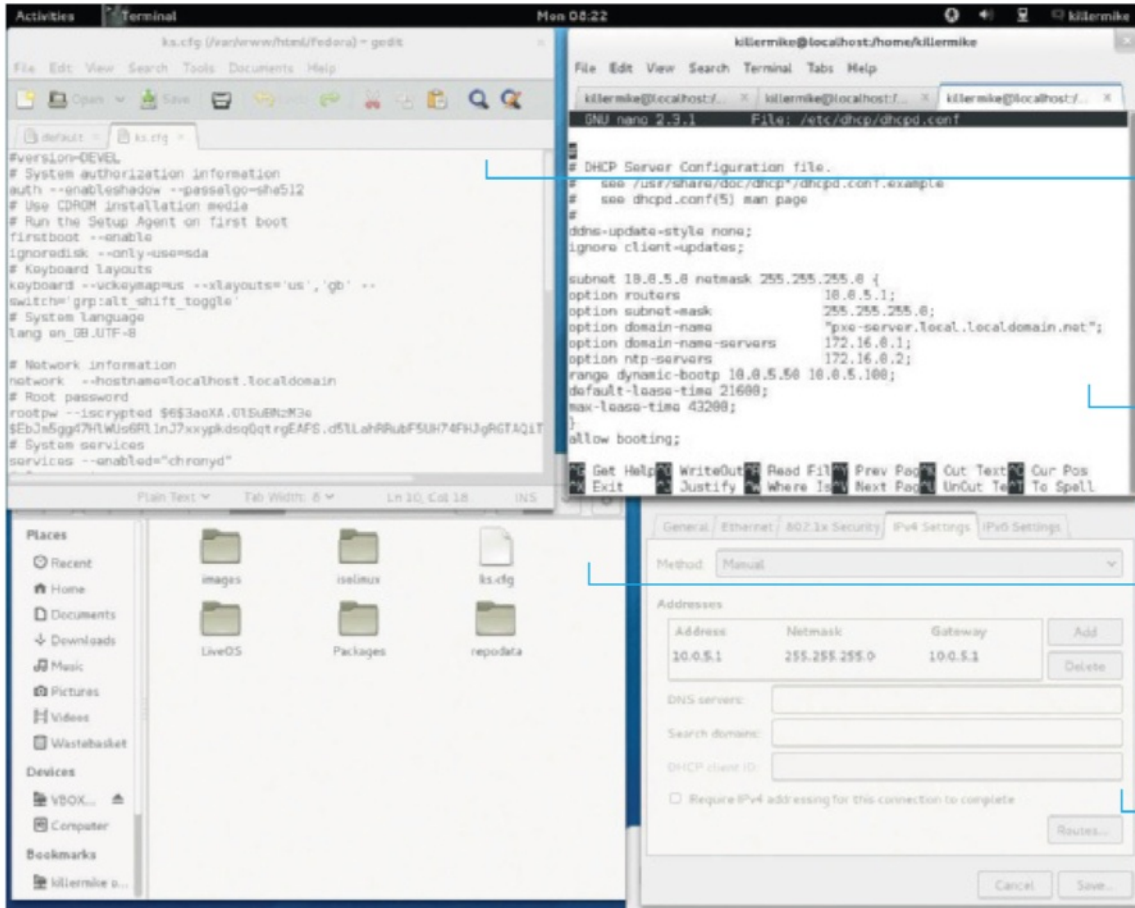


09 Turn on logging

Sometimes, rules don't actually do what you planned, but there are a number of tools for logging and manipulating rules. It's wise to be able to review the logs to see exactly what's going on. To turn logs on, simply go back into the Rules menu, find the rule that you think may be problematic, and tick the 'Log this rule' box. Don't forget that rules are evaluated on a first-match basis; so, for example, having the drop all rule before the rule trying to be tested would mean the rule would never get evaluated.

Backing up is also an important exercise and very simple to execute. Go to the menu, select Diagnostics>Backup/Restore. The options on this page are simple enough. It is recommended to tick the box to encrypt the backups. Give it a good password that you will remember. We also suggest you leave the box 'Do not backup RRD data' selected. This is just performance data and isn't really needed day-to-day.

Should the firewall ever need rebuilding from scratch, you will have to redo the steps right up until you have the GUI. The Restore menu, found in the Diagnostics menu, has the tickbox to restore from backup, but also the option to only restore parts, such as the rule base.



The Kickstart file provides all of the answers for the installer so that it doesn't have to prompt the user

A boot server needs to run a properly configured DHCP server so that connected machines boot from it

We'll place the contents of an ISO DVD into a folder that is served over HTTP

In this example, we shall set up a boot server that has two network cards – one for internet access and one for installation targets

Deploy Fedora over a network

Learn how to install Fedora to an entire LAN



Resources

- Working Fedora box
- Two network adaptors
- Fedora installation DVD ISO
- Network of at least two machines

Installing Linux on a single box is easy, but try extending that to a room, or even building, full of computers and you'll face a massive headache. To save you from running back and forth between all those computers, we'll show you how to set up an automated network install.

This project has two main stages. Firstly, a working boot server must be established. Secondly, a Kickstart file must be created to satisfy the installer and ensure that it does not require any interaction from the administrator.

Some solutions of this type favour making a bootable respin of the installation medium, but

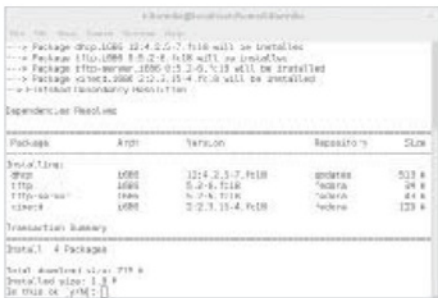
the problem with that method is that it becomes extremely tricky to make changes to the setup on installation day. So instead we're going to look at an approach that works from within a normal Linux installation, Fedora in this case. If you need it to be portable, no problem – just install Linux to a flash drive and work from that.

It ought to go without saying, but be a bit careful when connecting the server up to the switch/router of the clients. When fully configured, this machine will happily wipe and configure anything with which it comes into contact.



01 Install Linux

These examples use Fedora Linux, but any Red Hat-derived distro should work. If you install to a removable medium, make sure that you have enough free space to make a copy of the installation DVD ISO. 16GB of free space is a sensible minimum.



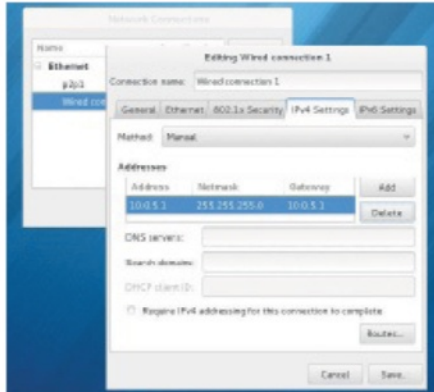
02 Add packages

Make yourself root (type `su` into a terminal), then use YUM to add to extra packages with `yum install dhcp tftp tftp-server xinetd`. You're going to be working as root, so, if you need to launch a GUI tool such as gedit, use `sudo gedit` (as root).



03 Set up network

These examples use a machine with two network adaptors – one for connection to the outside world, and one to connect to the machines that need to boot from it. The second network card probably doesn't have an IP address assigned yet, so we'll set this via the GUI.



04 Configure second adaptor

Right-click on the network icon and select Edit Connections.... Now locate the second adaptor, click on Edit and select the IPv4 Settings tab. Change the method from Automatic (DHCP) to Manual. Add a static IP address for your adaptor. For example, if your first adaptor is on 10.0.1.1, adding the second adaptor with an address of 10.0.5.1 and a netmask of 255.255.255.0 will give you space to connect up to 255 machines to the boot server.



05 Obtain dhcpd.conf

DHCP assigns IP addresses and starts the boot process on clients. Visit the official Fedora documentation (tinyurl.com/luad-dhcp) site to cut and paste an example DHCP configuration for a boot server. Load the existing file (`/etc/dhcp/dhcpd.conf`) into a text editor.

06 Modify dhcpd.conf

Modify the example `dhcpd.conf` to match your network. The subnet for our example would be changed to 10.0.5.0, and routers is the same address as your second network adaptor. Setting `range dynamic-bootp` to 10.0.5.50 10.0.5.100; gives space for 50 machines. The parameter `next-server` should be set to the same address as your second network adaptor. Change filename "linux-install/



`pxelinux.0`"; to filename "pxelinux.0"; Save the file and then start the server with `systemctl start dhcpd.service`.



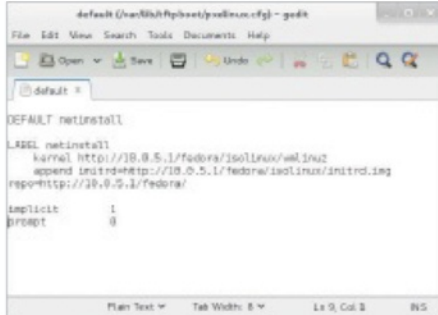
07 Configure TFTP

TFTP is a basic file transfer protocol that the NIC firmware uses to fetch the bootloader. Load `/etc/xinetd.d/tftp` and change the line `disable = yes` so that it reads `disable = no`. TFTP is managed by `xinetd`, so start with `systemctl start xinetd.service`.



08 Obtain PXELINUX

PXELINUX is the Linux bootloader that works over Ethernet. To get it, install `SYSINUX` with `yum install syslinux`. The file we need is `pxelinux.0`. Copy it to the TFTP folder with `cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/`. Type `mkdir pxelinux.cfg` to create the configuration directory.

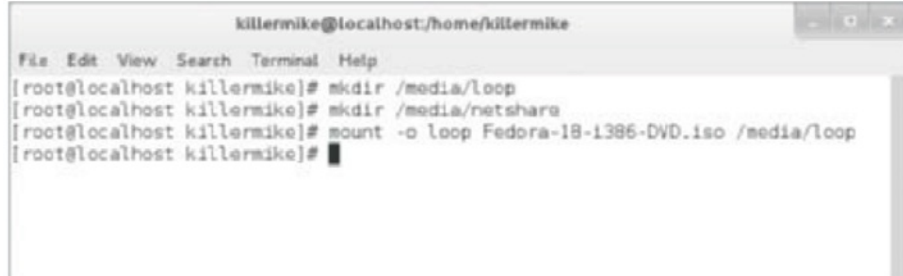


09 Configure PXELINUX

Type `cd /var/lib/tftpboot/`. Make a directory with `mkdir pxelinux.cfg`. Within this directory, create a text file called `default`. Add the following lines **DEFAULT netinstall**, **LABEL netinstall**, **kernel vmlinuz**, **append initrd=initrd.img** **repo=http://10.0.5.1/fedora/**, **implicit 1** and **prompt 1** so that it looks like the picture above. If you're feeling adventurous, try adding **prompt 0** so that clients won't wait for user confirmation before beginning the install. Be careful with that option!

10 Configure web server

Add Apache 2 with the `yum install httpd` command, and start it with `systemctl start httpd.service`. Test that it is up and running by navigating a web browser to `http://10.0.5.1`. If everything's working, you should see the Apache startup page.



11 Extract the ISO image

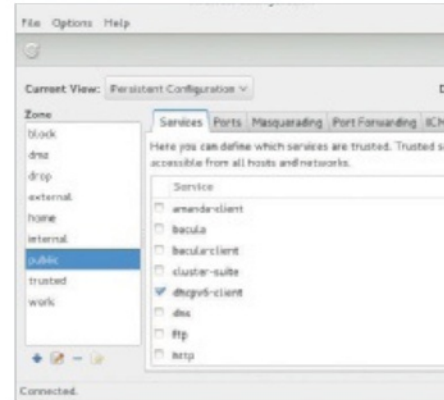
Create two directories: `/var/www/fedora` and `/media/loop`. Copy the Fedora DVD ISO image to the current directory and type `mount -o loop -t iso9660 [path to ISO] /media/loop`. Use `rsync` to copy the files: `rsync -v -a -H /media/loop/ /media/var/www/fedora`.



12 Copy vmlinuz and initrd.img

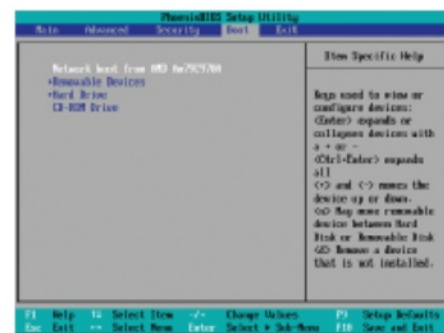
Enter the directory that TFTP can see with `cd /var/lib/tftpboot/`. Execute

`wget http://10.0.5.1/fedora/isolinux/vmlinuz`. Copying the file like this makes a good test that the server is working. Now retrieve `initrd.img` from the same directory.



13 Firewall

Open the Firewall configuration application. Select persistent configuration. Add `http`, `https`, `tftp` and `tftpcient` to the list of trusted services. Select `Reload firewalld` from the Options menu.



14 Ready the clients

Enter the BIOS setup screen of a client PC and make sure that the boot order specifies network booting as the priority. When carrying out the installation, you will disconnect the router/switch from the internet and connect it to the boot server instead.

“A connected machine should now boot from the server”

```

Intel UNDI, PXE-2.1
PXE Software Copyright (C) 1997-2000 Intel Corporation
Copyright (C) 2010 Oracle Corporation

CLIENT MAC ADDR: 08 00 27 33 19 CB  GUID: 6C45D30C-FBA4-4B90-ADDA-E192B1Z7FBA1
CLIENT IP: 10.0.5.52  MASK: 255.255.255.0  DHCP IP: 10.0.5.1
GATEWAY IP: 10.0.5.1

PXELINUX 4.05 2011-12-09  Copyright (C) 1994-2011 H. Peter Anvin et al
IPXE entry point found (we hope) at 9DDC:0104 via plan A
UNDI code segment at 9DDC len 199E
UNDI data segment at 9C59 len 1030
Getting cached packet 01 02 03
My IP address seems to be 00006534 10.0.5.52
ip=10.0.5.52:10.0.5.1:10.0.5.1:255.255.255.0
BOOTIP=01-00-00-27-33-19-cb
SYSBIOS=6ca5d30c-fba4-4b90-adda-e192b1z7fda1
TFTP prefix: /
Trying to load: pxelinux.cfg/default
Loading vmlinuz...
    
```



15 Testing 1

A connected machine should now boot from the server. If it doesn't work, there are some things you can try. Does the machine that is attempting to boot indicate that it has been assigned an IP address? If not, the problem lies with DHCPD on the server, so recheck /etc/dhcp/dhcpd.conf.

```

killermike@localhost:~/killermike
File Edit View Search Terminal Tabs Help

killermike@localhost:~/... killermike@localhost:~/home... killermike@localhost:~/home...

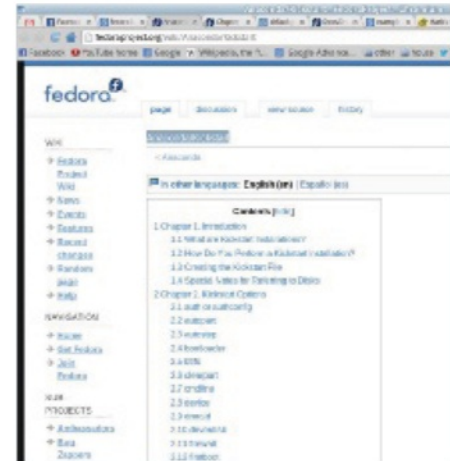
[killermike@localhost ~]$ su
Password:
[root@localhost killermike]# systemctl enable httpd.service
ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.wants/httpd.service'
[root@localhost killermike]# systemctl enable dhcpd.service
ln -s '/usr/lib/systemd/system/dhcpd.service' '/etc/systemd/system/multi-user.target.wants/dhcpd.service'
[root@localhost killermike]# systemctl enable xinetd.service
[root@localhost killermike]#
    
```

16 Testing 2

If the client tries but fails to load a file called pxlinux.0, it is communicating with DHCP, but TFTP may not be working. Try using the command `tftp 10.5.0.1 -c get pxlinux.0` on the server. If this retrieves the file, try executing it again on another machine. If the installer begins to boot, can find pxlinux.0, vmlinuz and initrd.img but stops at that point, try retrieving one of the files in /var/www/http/fedora/ manually by using the `wget 10.0.5.1/fedora/[name of file]` command.

17 Make services permanent

Control Fedora services with `systemctl [command] [service]`. The main commands you'll need are `start`, `enable` to make permanent, and `restart` when you make configuration changes. This project requires running `httpd.service`, `dhcpd.service` and `xinetd.service`.



18 Create Kickstart file

A Kickstart file supplies the installer with answers to avoid prompting the user. To begin, create a file called ks.cfg in /var/www/html/fedora/. Go to the official Fedora Anaconda/Kickstart page for a complete list of commands (tinyurl.com/luad-kickstart). When a Fedora system has been successfully installed, a (fully commented) Kickstart file is deposited in /root/anaconda-ks.cfg; this makes a good starting point for building your own. If you installed via the ISO, remove the line that sets install type to CDROM.



19 Configure Kickstart file

You must specify a root password using the `rootpw` command to avoid a prompt. You may want the installer to erase all partitions (or preserve some). The `upgrade` command causes the installer to upgrade the targets rather than carry out a fresh install.

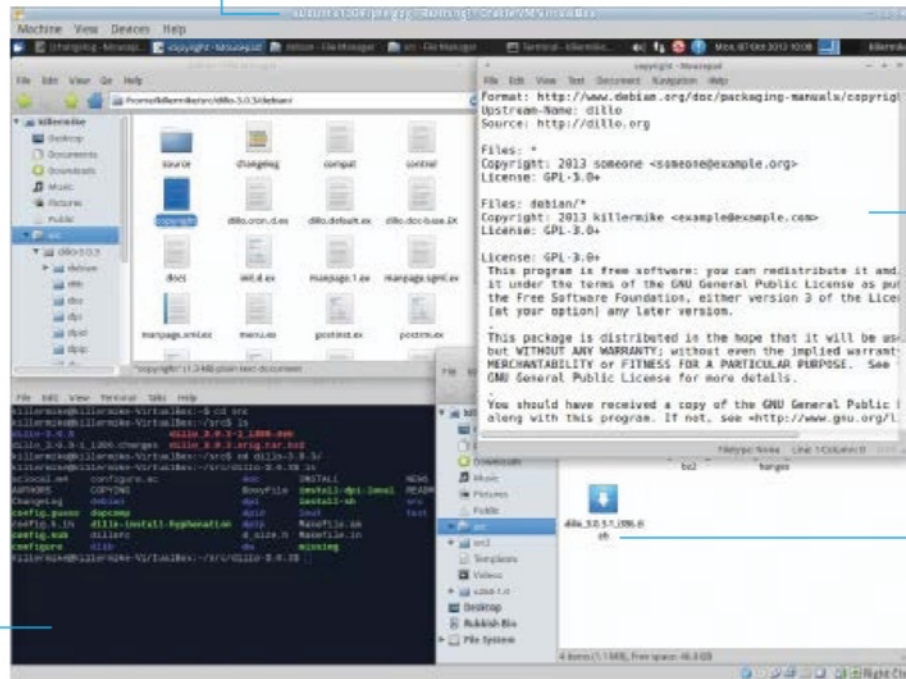
Save your custom Kickstart file as /var/www/html/fedora/ks.cfg and then add `ks=http://10.0.5.1/fedora/ks.cfg` to the append line in your default file. By default, the finished target machines will begin in first-run mode and ask the user to specify details such as username and password on the first run.

We recommend that you carry out this tutorial inside a virtual machine rather than on real hardware

Whether building RPMs or Debian packages, the configuration consists of editing some text files

The actual building of packages takes place from the command line, as does much of the setup

The finished product, a DEB file that can be installed on Debian-derived distros



Make your own DEB and RPM packages

We'll show you how to manufacture the two most common types of Linux package for software distribution so you can become your own package maintainer

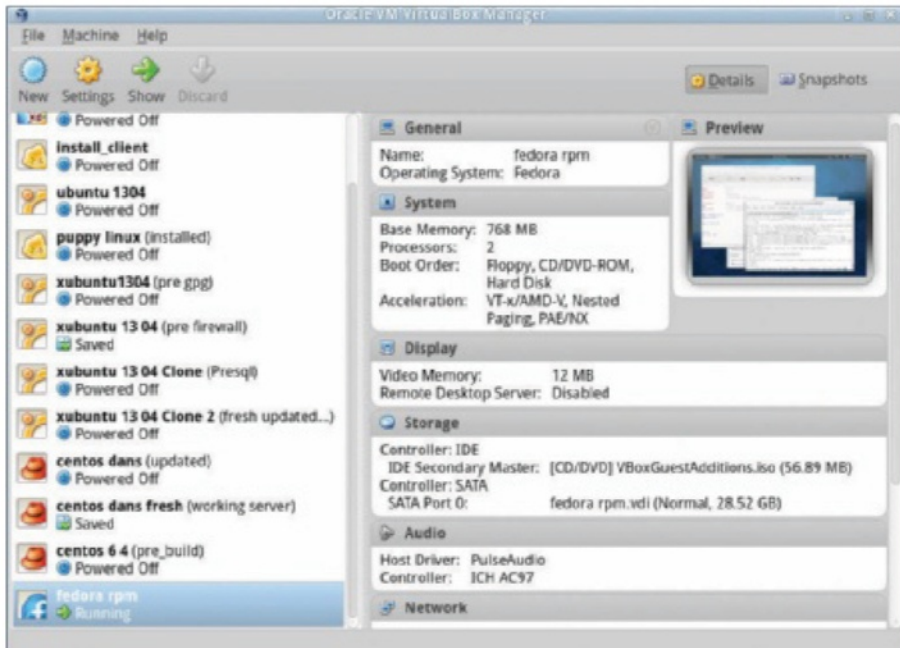
We're going to take you through the process of making software packages for the two most popular packing systems, DEB and RPM. You can use these techniques to package your own software or even to become a package maintainer for software projects that you feel are being overlooked.

We'll start with a guide to building DEB (.deb) files for Debian-derived distributions – we're using Xubuntu as our base for that. Following that, we'll detail the methods needed for the creation of RPM packages for use on Red Hat-derived distributions, and we'll use Fedora for

that. You can often create a package on one distribution and then install it on a related one (Ubuntu>Debian, for example), but it might be worth testing it yourself, if this is crucial.

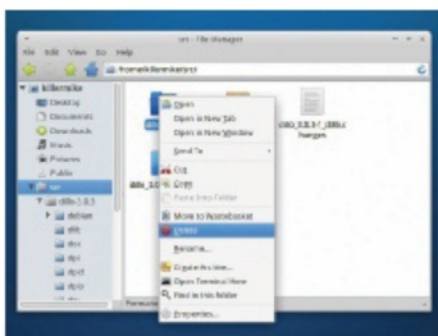
As for the software, we're going to use Dillo, a lightweight web browser, as an example package to build from source code. As is often the case when building from source, you may have to look around on the web for solutions if the build doesn't go as it should. For example, in the case of Dillo 3.0.3, we had to add 'LIBS=-lX11' to the front of the build commands to get it work, due to an oversight in the source code archive.

Resources Ubuntu & Fedora installation (or VM)



01 Employ a virtual machine

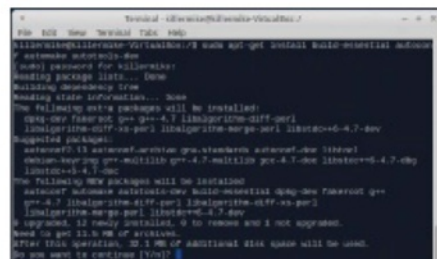
Using a virtualiser such as VirtualBox or VMware is often the best approach to building packages for other systems. For one thing, it allows you to maintain a relatively clean, reference installation that is comparable to a setup that other people are likely to be running. This also means that you can keep a selection of target environments, using a different distributions. In addition, most virtualisation products allow the emulation of different architectures, and this can even extend to running a 64-bit OS on a 32-bit platform, although performance will suffer.



02 Starting from scratch

If things go wrong, with Ubuntu or Fedora, it is perfectly safe to simply delete the source directory and start again. Note that the Debian tools do alter the source archive, so you'll have to start with a fresh copy.

Part 1: Debian



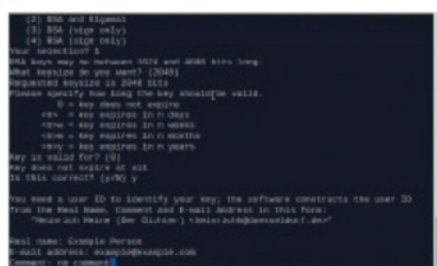
03 Install build environment

We'll start by installing most of the tools that we need for to make software from source code. Type:

```
sudo apt-get install build-essential
autoconf automake autotools-dev
```

Now we have to install tools that are used for handling DEB packages. Do this with the following command...

```
sudo apt-get install dh-make
debhelper devscripts fakeroot xutils
lintian pbuilder
```



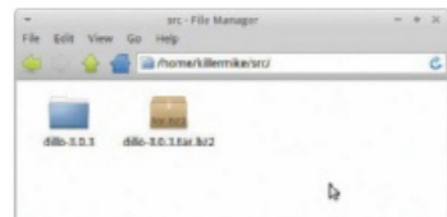
04 Create a GPG key

If you haven't created a public GPG key in the past, you must create one now so you can sign packages. Start by typing `gpg --gen-key`. Accept the default settings, and fill in your details. Make a note of these, as we need an exact match later. Following this, type `ls ~/.gnupg` to make sure the new key exists (it's `firstname.lastname.gpg`). Create a public key from this with: `gpg -a --output ~/.gnupg/[your key].gpg --export '[your name]'`. Import this with `gpg --import ~/.gnupg/[your key].gpg`



05 Fetch package

In this example, we're going to fetch and build the latest version of the Dillo web browser. Navigate to the Dillo website (www.dillo.org) and download the most recent .tar.bz tarball. Create a directory for source code with `mkdir ~/src` and move the archive into it.



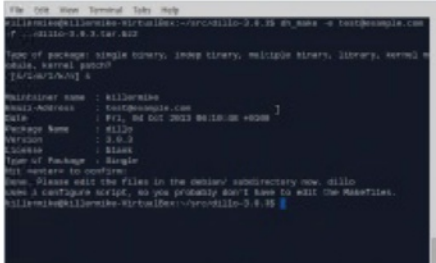
06 Unpack the archive

Unpack the archive with `tar -xjvf [archive name].tar.bz2`. Note the naming convention of the directory (package name-version) is crucial, and fortunately Dillo complies with this. It's also crucial that the source archive is one level above the source directory.

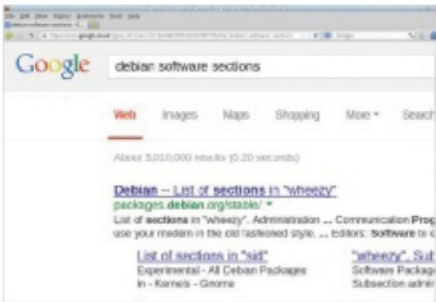
07 Add Debian compliance

Move into the directory that we have just unpacked with `cd`. `dh-make` is a script that takes care of adding the configuration file and directory structure that we need; it's part of the debhelper suite that we added earlier.

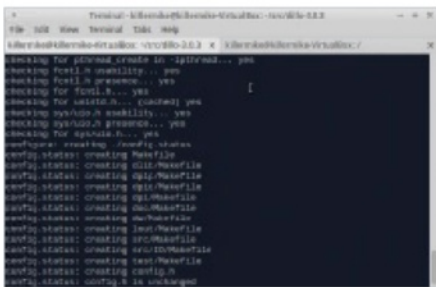
```
dh-make -e [your email address] -c
licence -f ../[source archive]
```



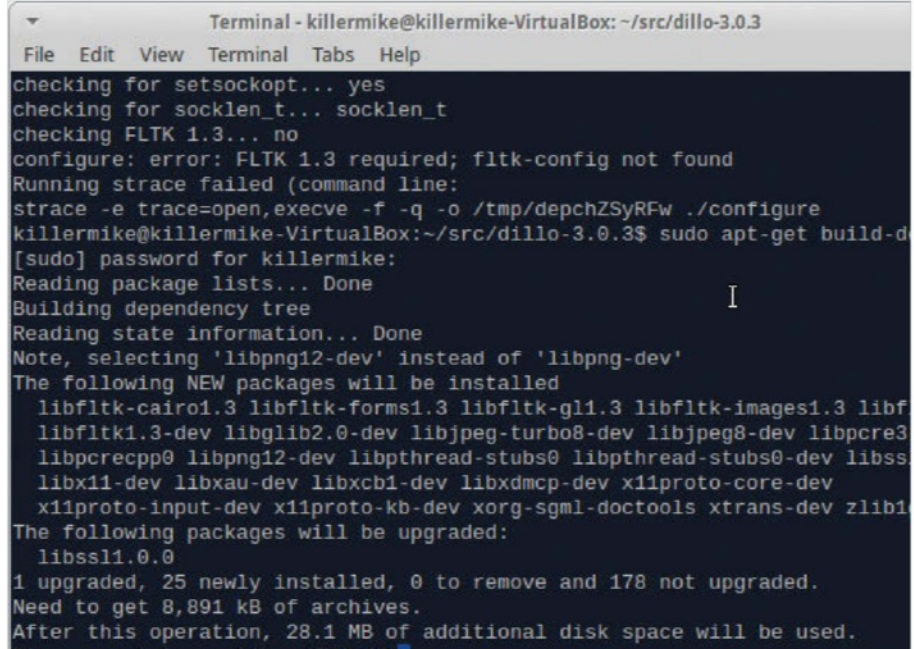
In our example, the command line is:
`dh_make -c gpl3 -e example@example.com -f ../dillo-3.0.3.tar.bz2`
 When prompted, select single binary. The helper script should have created a directory called Debian within the source code directory.



08 Open the control file
 Open the file control in the debian subdirectory in a text editor. Fill in the homepage section (Google for complete list of Debian software sections) and description fields of this file.



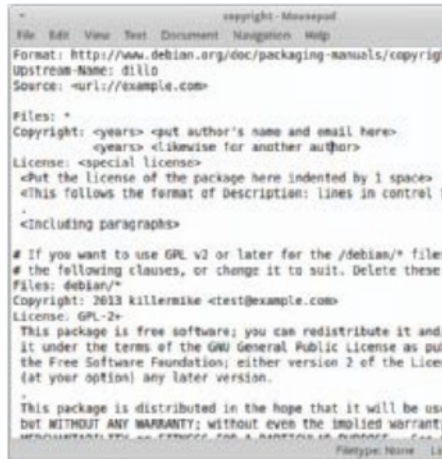
09 Discover dependencies
 You can discover the dependencies needed to run the software in the package by moving into the source directory and typing `dpkg-depcheck -d ./configure` into a terminal. This may produce errors that indicate a package needed in order to build the software is missing. You can discover these packages by typing `sudo apt-get build-dep [name of package]`, and this should help if there is some support for this software in the repository of the distribution. If not, you'll have to repeatedly



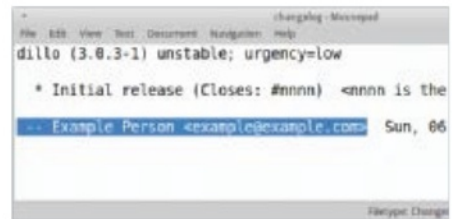
“If everything is set up correctly, we can finally build the DEB”

run `dpkg-depcheck -d ./configure` and add packages manually by typing `sudo apt-get install [name of package]`.

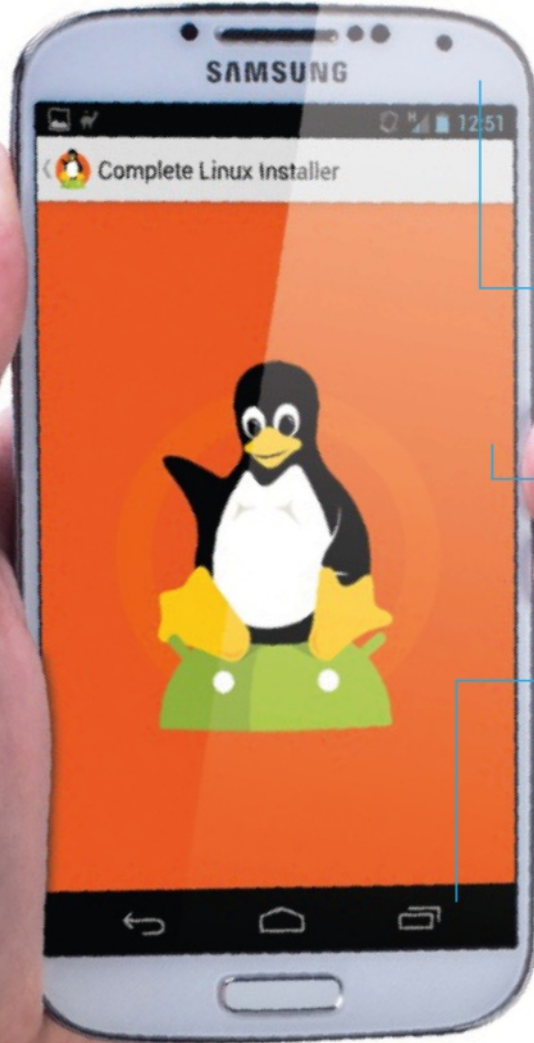
10 Add dependencies to the control file
 When the command from the previous step has completed, it should present you with a list under the packages needed heading. Add this list of dependencies to the depends: section of the control file. Each item on the list must be separated by a comma and a space.



11 Edit the copyright file
 Try to complete this step as comprehensively as you can, and don't skip it. Source: is usually the homepage of the project. Within the Files: * section, replace the copyright information with the names of the authors of the project. You can see the required format for this by examining the Files: debian/* section, which should have your details in it. You may have to do a bit of detective work to find the information you need. Look for files such as AUTHORS and COPYING within the source directory.



12 Edit the changelog file
 Open the changelog file and make sure that the name and email address match those that you entered when creating your GPG key. Typically, the helper script may have added your username rather than your real name to the file. As with the copyright file, don't skip over



Take your powerful pocket computer and turn it into a true Linux system

Linux on Android lets you create virtual Linux machines on your rooted Android device

Connect through to your phone via a VNC client on your PC

Install a Linux distro on Android

Put Linux on your Android device using the LinuxonAndroid app and keep a portable open source computer in your pocket with our tutorial

Resources

A compatible, rooted Android device

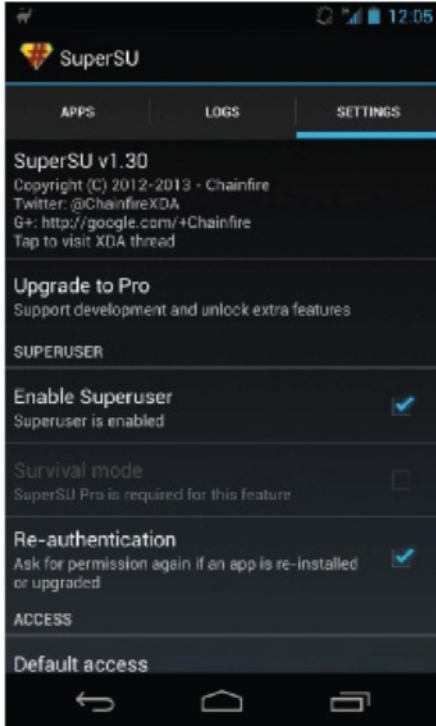
LinuxonAndroid app: linuxonandroid.org/

Ever since Android was launched, people have been trying to make the most out of these pocketable devices which are more and more able to double up as portable computers. While Android was built upon Linux, it's not truly the same kernel any more and with the proliferation of ARM-compatible distros, the prospect of putting Linux on an Android device to unlock its full potential is very attractive.

With LinuxonAndroid (LoA), not only can you get a Linux distro on an Android device, but you can also keep the original OS on there to use as normal. LoA uses a virtual machine to install a

selection of supported distros to your phone. This requires root access to your phone and while it doesn't support every single Android device, there are over 500 supported. This includes the Google Nexus 4, 7 and 10, and the Samsung Galaxy S III and 4.

Currently, the distribution selection is quite small – Ubuntu 10.04, Ubuntu 12.04, BackTrack Linux and Debian are available to install right now. There is work going on to get support for Fedora, Arch and more, and this tutorial will work for them as well when they are eventually released.



01 Root your phone
Most Android devices have a different root method, some easier than others. There are some common elements between them – making sure to back up, unlocking the bootloader, possibly flashing a recovery. The Nexus 4 can be one of the harder devices to unlock, so we will focus on that.

02 USB debugging
You'll need to turn on USB debugging from the developer menu on all devices for this. Jelly Bean or newer Android builds have developer options hidden by default, but you can activate them. Go to 'About phone/tablet' and tap 'Build number' seven times to unlock it, then check 'USB debugging'.

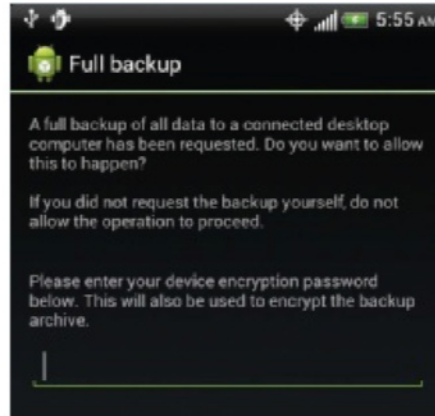


03 Download root tools
The DownloadAndroidRom.com site has a number of compressed files with rooting tools for a variety of devices. Download the

correct one for your phone or tablet, extract the files, and then navigate to the new folder in the terminal and type:

```
$ sudo chmod 755 *
```

Connect your device to your PC now.



04 Backup
The Nexus 4 needs to be wiped to unlock the bootloader, so the tools come with an ADB backup solution. In the same terminal you just used to make the files executable, run the following command:

```
$ ./adb-linux backup -apk -all -f backup.ab
```

Click 'Back up my data' on the device. This will not save data on the internal storage, though.

05 Unlock bootloader
Turn off your device and reboot into the bootloader. On Nexus devices, hold volume-down and power buttons to do this. Once it's turned on, you will be presented with the bootloader screen. Now type in your terminal:

```
$ ./adb-linux oem unlock
```

Select yes using volume keys and power button.



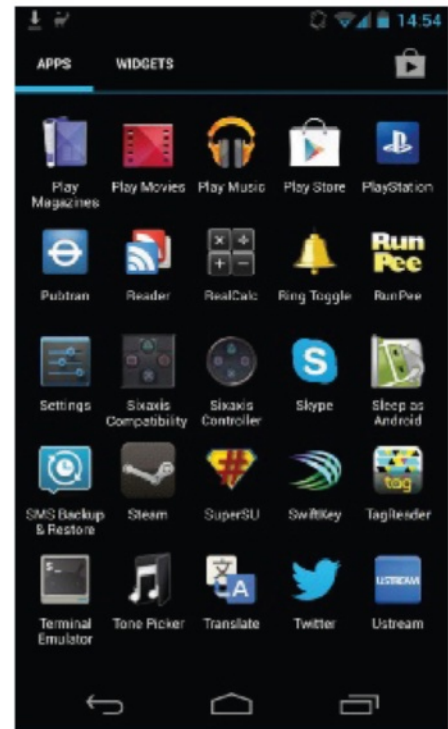
06 Superuser files
Turn your device back on and go through the setup process – skip it for now. You'll need to

put CWM-SuperSU-v0.99.zip onto your Nexus storage, either via AirDroid or by downloading it straight from DownloadAndroidRom.com through the browser.

07 Recovery mode
Reboot your phone back into the bootloader and make sure the USB cable is still plugged in. In the terminal, type:

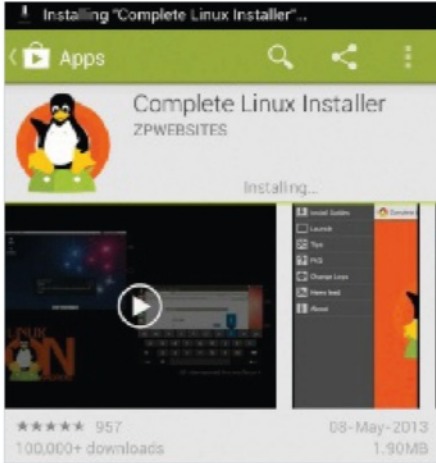
```
$ ./fastboot-linux flash recovery recovery-clockwork-touch-6.0.2.3-mako.img
```

Use the volume buttons to select Recovery mode and press power.



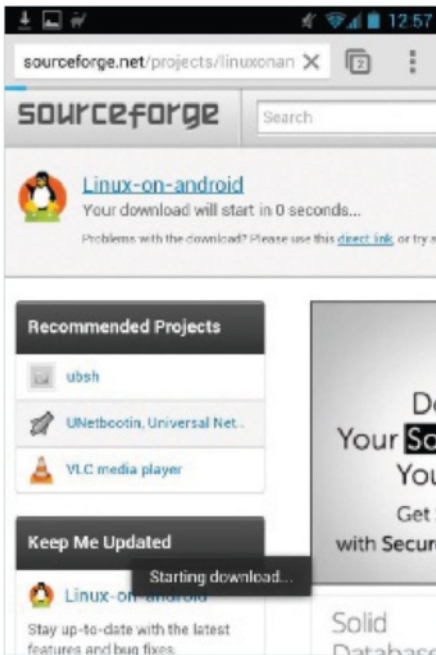
08 Flash recovery
Select 'Install zip from sdcard', followed by 'zip from sdcard' and then 0/. Choose CWM-SuperSU-v0.99.zip and then hit Yes. After that's done, select Reboot; if asked if you want to disable Flash Recovery, select yes. Afterwards, you should find the SuperSU app in your device's app drawer.

09 Rooted
Your Nexus 4 should be rooted – open the SuperSU app to run some binaries and make some changes. Check out AndroidRoot.org to finish rooting other devices on Linux, with the Galaxy S phones being fairly easy and not requiring a wipe.



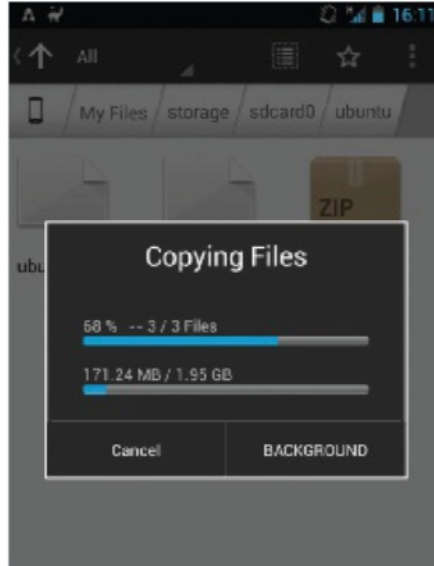
10 Install apps

Now you're all rooted and superuser'd, install three apps from the Play Store: Complete Linux Installer, androidVNC and Terminal Emulator. You can find links to these in the guide for Complete Linux Installer's distros. Install these after rooting, not before.



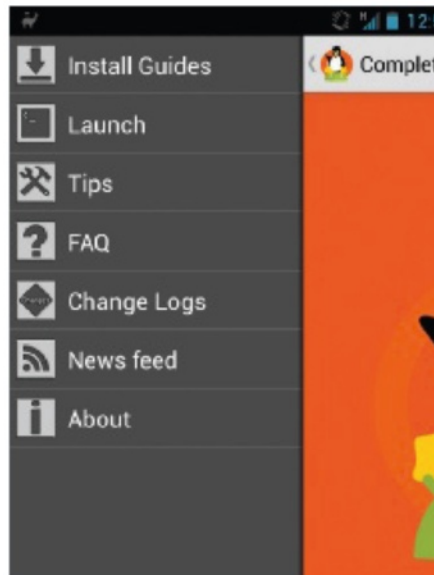
11 Download image

You can download the images from the LinuxonAndroid SourceForge project page, or via the distro guides on the app. The Ubuntu and Debian images come in three sizes: Core for a command-line interface; Small for LXDE; and Full for Unity. Each is larger than the last, and heavier, so choose carefully.



12 Unzip image

Use an Android file explorer such as Astro File Manager and navigate to the zip files on your device. Move them to a folder named after the distro with lower-case first letter (ubuntu, debian, backtrack) and extract them into the folder. This will have the image and md5 sum.



13 Launch distro

Go back to the Complex Linux Installer app and open the sidebar. Select the correct distro from the drop-down list and click Launch. This will open a terminal and check the md5 hash before guiding you through some extra setup processes.



14 First-time setup

It will run through some setup options with you. Firstly enter a password for the image; make sure to use the name of the distro. You'll be asked if you want to turn on VNC and SSH servers – VNC is mandatory, but SSH is optional. Set the resolution to your device's, save the profile and wait until 'root@localhost' shows up



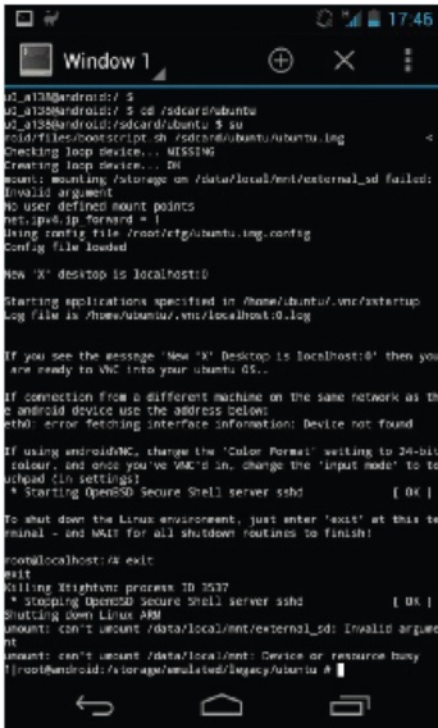
15 androidVNC

Now you need to connect to the VNC server you just turned on. To do this, go to androidVNC, set the password to the name of the distro with the first letter lower-case, put localhost in the address, and set the port to 5900. Finally, press Connect and you will be put into the desktop of the selected, running image.



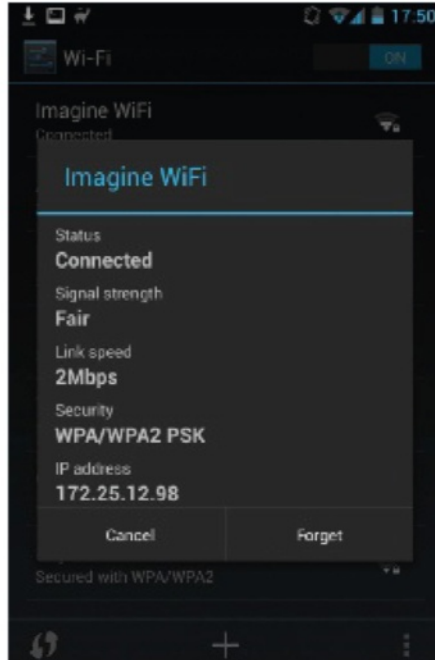
16 Mousing

If you put a larger resolution than the screen can show, or zoom in, you can move around the desktop by swiping your finger. The mouse pointer reacts as you touch the screen, rather than following your finger, so you'll need to hold down your thumb or finger to drag files or directories.



17 Shut down

Don't just turn off the terminal emulator when exiting the distro – this will not turn off the distro, and you'll need to then reboot your phone to get it to properly boot again. Instead, go back to the terminal emulator and type 'exit'. Wait until it's fully shut down, then exit the window.



18 Phone IP

You can connect to the Linux session running on your phone via a remote desktop client on your PC. The first thing you'll need to do is make sure your phone is connected to the same Wi-Fi network as your PC. Press the wireless connection in the Wi-Fi menu and note down your local IP.



19 VNC on PC

Open a VNC client on your PC and click on a new connection. Ensuring the protocol is set to VNC, enter the IP address of your device followed by :5900 to specify the port. Use the distro password for the username and password, set colour depth to 24-bit if needed, and connect.

20 Install package manager

You'll need to install apps from the terminal, and it's faster to do it from the phone or tablet's terminal emulator. If you want to be able to install with a graphical package manager, use: `# apt-get install synaptic` ...to install the Synaptic Package Manager.

“You can download the images from the LinuxonAndroid SourceForge project page”

21 App limitations

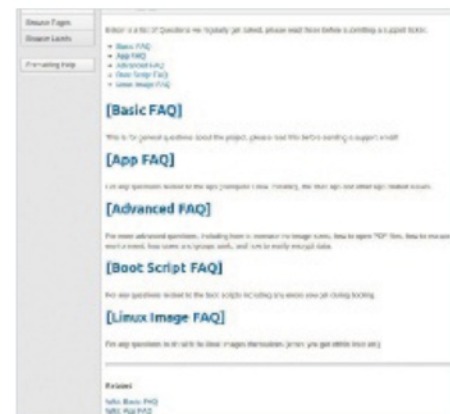
As we're using the ARM version of the distros, there won't be the same number of apps available as for the x86 versions. For example, Wine on ARM is still under heavy development, and is not recommended. Make sure to download ARM binaries as well when installing that way.

22 Uses

Having a portable Linux distro on your phone can be great for many things. With the VNC capabilities, you can use it as a portable desktop, much like Ubuntu for Android. The storage on the device is permanent, so it will save any files and cookies for access to cloud storage.

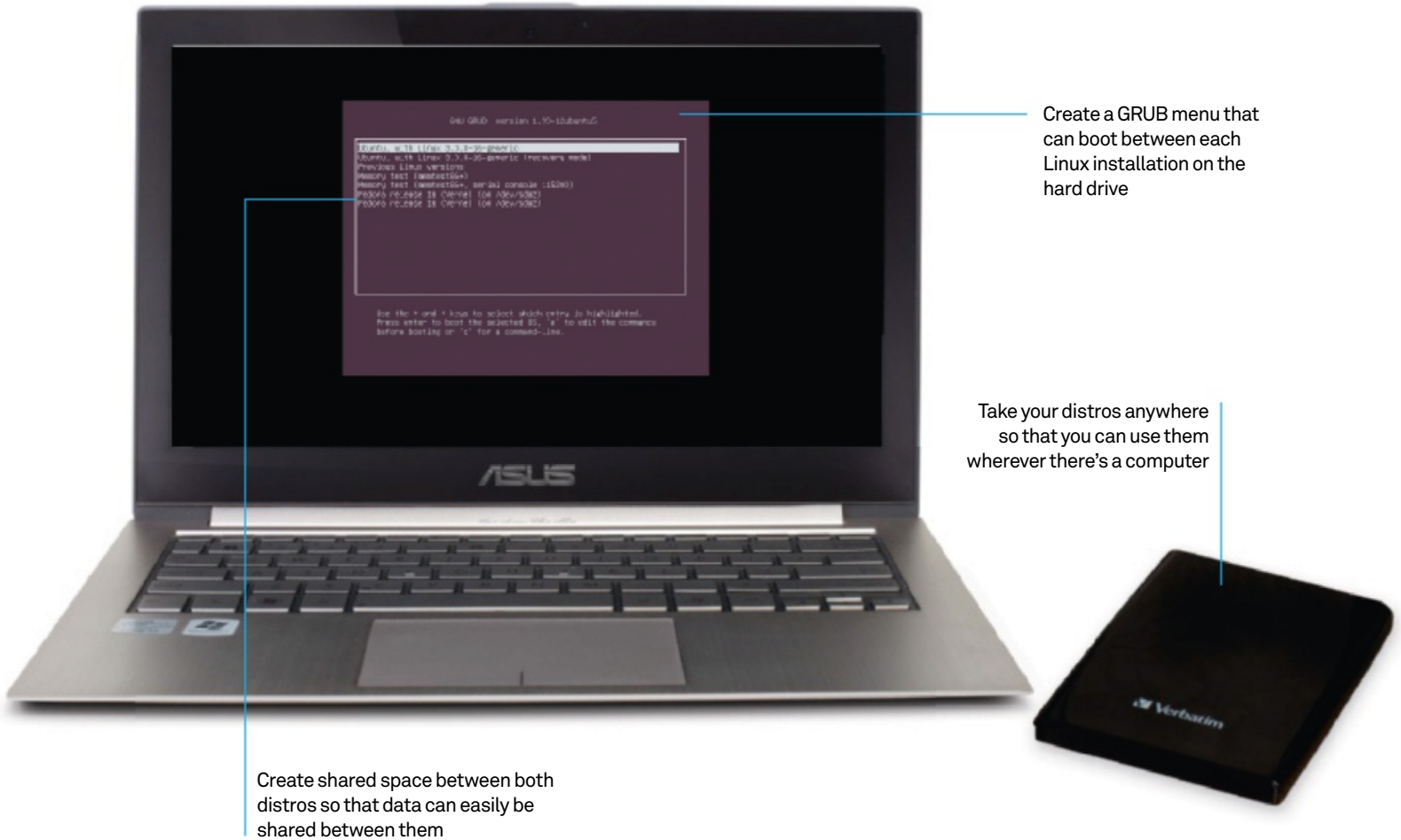
23 More Uses

Alternatively, in command-line mode you can have a low-power, portable and small file or web server. Thanks to BackTrack being available, you also have a portable network sniffer and vulnerability checker.



24 More to do

If you're still looking to do more with your device, the LinuxonAndroid website has a number of advanced tutorials on how to expand the storage, run apps through the Android activities manager and much more. Now you truly have a PC in the palm of your hand.



Create a GRUB menu that can boot between each Linux installation on the hard drive

Take your distros anywhere so that you can use them wherever there's a computer

Create shared space between both distros so that data can easily be shared between them

Dual-boot from an external hard drive

Want to carry around a multi-booting hard drive you can connect to any computer? Then this easy-to-follow tutorial is exactly what you need

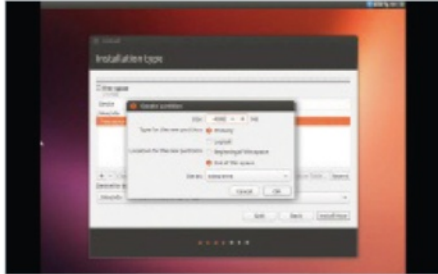
Resources

A USB-powered external hard drive with at least 75GB of space

Carrying around a USB stick with a preloaded live distro is very useful. It can be used as an emergency distro (for a PC experiencing some troubles), diagnosis tools, a portable distro for yourself and much more. However, when it comes to using it as a portable distro, there are some limitations. Not all live distros allow you to install extra software permanently, and even then they may have a limited repository of software that can be installed in a live environment.

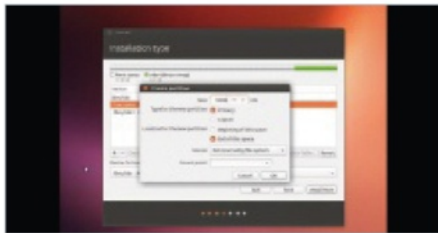
This isn't a problem if you have enough storage space, though, such as on a portable, USB-powered external hard drive. You can install distros to them much as you would on a normal hard drive. However, there are some changes you'll need to make if you plan to dual-boot from this hard drive on any machine.

Here we show you how to partition, install and run two distros from an external drive. We'll use Fedora and Ubuntu in our example, as they are two of the most common distros.



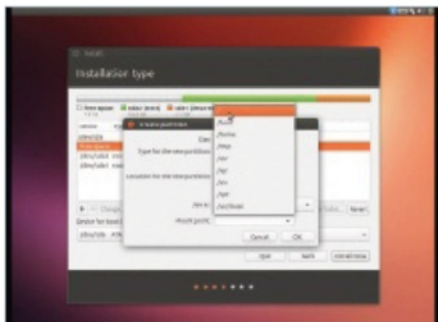
01 Swap partition

In our example, we'll start with Ubuntu. Plug in the USB drive and boot into Ubuntu. Make sure you've saved any important data on the external drive. The first thing you'll need to do is create a swap partition – make it 4GB and put it towards the end of the hard drive.



02 Shared space

You can create some shared space for both distros to use, either independently or as a shared home directory. We want to save 10-15GB for each distro, so keep that in mind while creating it. Use ext3 or ext4 as the file system, or NTFS if you want it to be cross-platform.

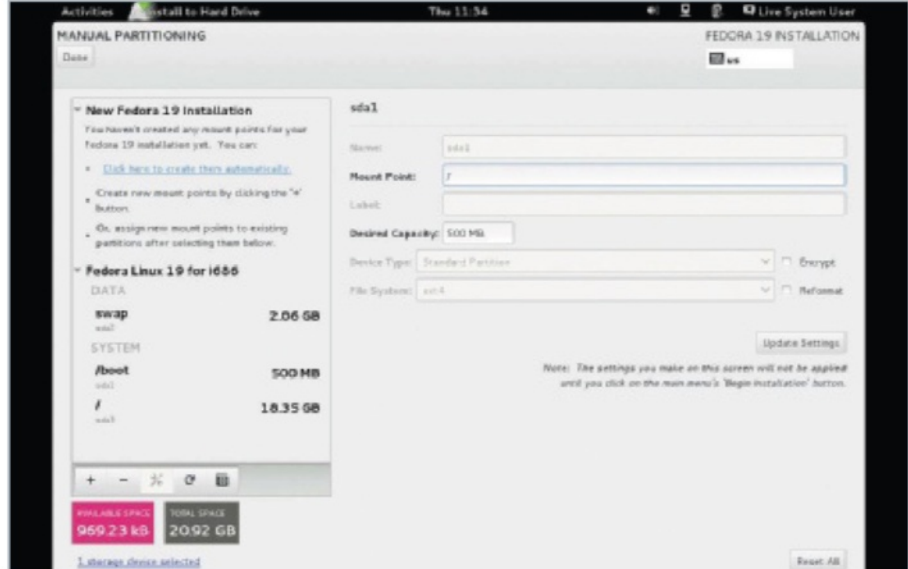


03 Ubuntu root

Create the 10-15GB space for Ubuntu and set it to be root by selecting '/' in the menu. It'll automatically mount our swap partition when booting into Ubuntu/Fedora in the future. Carry on with the installation instructions.

04 Fedora root

Put in the Fedora disc with the USB hard drive still plugged in and boot up. Choose the external drive from the list of disks in



Installation Destination, create a custom partition in the space remaining as ext3 or 4, and set the mount point to '/'. Install as normal.

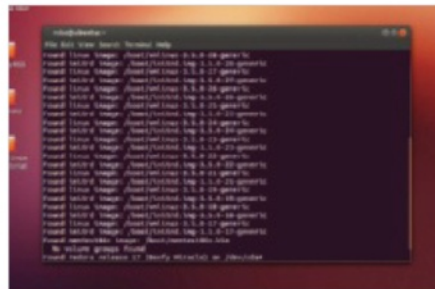
05 GRUB reinstall

Boot back into the Ubuntu live disc and make a note of what `fdisk -l` labels the Ubuntu boot partition on the external drive as. Mount it with:

```
$ sudo mount /dev/sdXY /mnt
```

And then reinstall GRUB 2 from Ubuntu with:

```
$ sudo grub-install --root-directory=/mnt /dev/sda
```



06 GRUB update

After rebooting, you'll be able to get back into Ubuntu on the external drive. Once there, mount the Fedora partition however you wish and run:

```
$ sudo update-grub
```

It will automatically detect the Fedora install

and update the boot menu next time you boot from the hard drive.

07 Find fstab

Enter the command:

```
$ ls -l /dev/disk/by-uuid
```

...to find out the UUID of the partition for your shared storage on the hard drive. This should be called the same on both Ubuntu and Fedora. Go to the terminal and open fstab with your favourite text editor like so:

```
$ sudo nano /etc/fstab
```

08 Use fstab

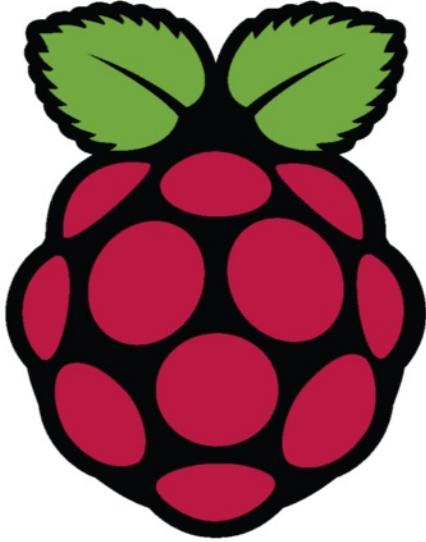
You'll need to add a new entry so that your shared space mounts every time you boot into one of the external drive distros. Enter into both fstabs something like:

```
UUID=XXXXX [mount point] ext4
errors=remount-ro 0 2
```

You can change ext4 to whatever other file system you used.

09 Go portable

You should now be fully set up and ready to take your portable Linux distributions everywhere, with a shared space between them, and the ability to boot between either of them without having to guess which partition is which.



You'll end up with a simple web interface you can access from anywhere

You can require authentication to access your web interface

We've made this to control a garage, but your imagination is the limit



Remotely control your Raspberry Pi

Take control of your Raspberry Pi from your smartphone, tablet or PC, from anywhere in the world

People are starting to do all kinds of things with their Raspberry Pi, like having it open and close a garage door, automate lighting and heating in the home, and so on. Even if you don't yet have a project like this, this tutorial will still serve as a useful introduction to writing web applications with Python.

We'll be using Arch Linux as the operating system for our project, because it is lightweight and we won't need a desktop environment.

This tutorial assumes that you have flashed the latest Arch Linux ARM image (in our expert's case archlinux-hf-2013-05-14) to an SD card. If you haven't, the instructions

for flashing an image can be found at www.linuxuser.co.uk/tutorials/how-to-set-up-raspberry-pi/. You'll only need to go up to the step where you write the image to the SD card. You'll have to adapt the instructions slightly for using the Arch Linux image rather than the Debian one.

Our VPN server will be made up of the following software components:

- Base Arch Linux system
- The Apache web server
- Python as a web scripting language
- `sudo` to elevate permissions in the CGI script if necessary
- A Dynamic DNS daemon (`noip`) – software that runs in the background and points a domain name to your router's IP address, meaning that you can access your Raspberry Pi from anywhere using an easy-to-remember web address

Resources

A Raspberry Pi

Arch Linux www.raspberrypi.org/downloads

A second computer – for SSH and testing

01 Logging into Arch Linux

Connect the necessary cables to the Pi and wait for the Arch Linux login prompt. The login is root, and the password is also 'root'. We'll change the root password from the default later on.

02 Running a full system update

Arch Linux runs on a rolling release schedule, meaning that there are no version numbers and software is continually updated. The package manager in Arch Linux is called pacman. Use the command `pacman -Syu` to start a full system update. If for some reason the update fails, try running the same command again. Sadly, the Arch Linux ARM servers tend to be quite busy. There may be a lot of packages to update so it may take a while, especially because the Pi runs from an SD card.

03 Installing the required packages

Use the command `pacman -S noip apache python2 sudo` to install the required packages mentioned at the start of the article. Answer 'y' to any prompts you may encounter.

04 Investigating your network

We highly recommend assigning a static IP to your server Raspberry Pi rather than being handed one by your router, because then you'll always know where to find it on the network, which will be useful for accessing it remotely. You'll also need a static IP if you want to access the Raspberry Pi from the internet. We'll need to find out a couple of things about your current network setup before setting a static IP. You can use the commands `ip addr show dev eth0 | grep inet` and `ip route show | grep default` to do this. We are using grep to filter out any information we don't need, by only displaying lines containing 'inet' to get the IP address of the Pi, and 'default' to get the default gateway (the address you route through to get to the internet).

05 Investigating your network

Now that we have found out things about your network, such as your current IP address, the subnet mask and so on, we can

“You will need a static IP address if you want to access the Raspberry Pi from the internet”

set up a static IP address. To do this, we're going to create a new systemd service. Create the file `/etc/systemd/system/static-network.service` with the following contents (replacing our IP address and default gateway with ones from your network):

```
[Unit]
Description=Static IP Connectivity
Wants=network.target
Before=network.target
BindsTo=sys-subsystem-net-devices-eth0.device
After=sys-subsystem-net-devices-eth0.device
```

```
[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/sbin/ip link set dev eth0 up
ExecStart=/sbin/ip addr add 172.17.173.254/24 dev eth0
ExecStart=/sbin/ip route add default via 172.17.173.1
```

```
ExecStop=/sbin/ip addr flush dev eth0
ExecStop=/sbin/ip link set dev eth0 down
```

```
[Install]
WantedBy=multi-user.target
```

Save the changes, exit nano and then run the following commands to disable DHCP and enable the Ethernet interface and the bridge with a static IP permanently:

```
systemctl disable netctl-ifplugd@eth0.service
```

```
systemctl enable static-network.service
```

You can now restart the Pi with the command `reboot` for the changes to take effect.

After restarting, `/etc/resolv.conf` will have changed, because it was configured by DHCP and we are no longer using that, so you'll want to run the command `echo nameserver 172.17.173.1 > /etc/resolv.conf`. Once you've done this, you'll be able to resolve DNS addresses like `google.co.uk` to an IP address.

06 Logging in with SSH

Once the Pi has booted back up, open a terminal on your Linux computer and type `ssh root@[ip of your Pi]`. Answer yes, to say that you want to connect, and type in the root password, which will still be root. You are now logged in over SSH.

07 Changing the root password

Since we will probably be exposing an SSH login to the internet (we might as well as we're exposing a web server), it would be a very good idea to change the password to something much more secure. Type 'passwd', and then follow the on-screen instructions to change your password. Your SSH session will stay logged in, but you'll need to use the new password next time you log in. You may also want to change the contents of `/etc/hostname` to set the hostname to a self-identifying name, such as `remotepi` rather than the default: `alarmpi`. The change won't take place until after a restart.

08 Configuring Apache

Apache is one of the most popular web servers in the world. It could be argued that a slightly less memory-hungry web server should be used on the Raspberry Pi, but Apache is tried and tested, and there are only ever going to be a few users on a remote-control system such as this. A web hosting company called Mythic Beasts hosted a mirror server with Apache on a Raspberry Pi. It was online for seven months and two weeks before the SD card failed. During that time, it shipped about 1.5TB of traffic.

Common Gateway Interface (CGI) is a standard method for a web server to delegate

```
[root@alarmpi ~]# ip addr show dev eth0 | grep inet
inet 172.17.173.254/24 brd 172.17.173.255 scope global eth0
inet6 fe80::ba27:ebff:fef3:9016/64 scope link
[root@alarmpi ~]# ip route show | grep default
default via 172.17.173.1 dev eth0 metric 204
```

the generation of web content to executable files. In our case, the executable file will be a Python script.

We need to edit `/etc/httpd/conf/httpd.conf` to enable the execution of CGI scripts, and to treat any file ending in `.cgi` as a CGI script. This file is quite long. We'll be editing a section that begins with `<Directory "/srv/http">` about 200 lines into the file. You need to change the Options line from:

```
Options Indexes FollowSymLinks
to:
Options Indexes FollowSymLinks
ExecCGI
```

Then add the following line under the Options line you just changed:

```
AddHandler cgi-script .cgi
```

Finally, you want to go to the section below the one we just edited and change the line:

```
DirectoryIndex index.html
to:
```

```
DirectoryIndex index.html index.cgi
```

This will automatically run the `index.cgi` script we're going to create instead of displaying a list of files that are in the directory.

```
<IfModule dir_module>
  DirectoryIndex index.html index.
cgi
</IfModule>
```

09 Starting Apache

Use the following command to start Apache at boot:

```
systemctl enable httpd
```

Then start it immediately:

```
systemctl start httpd
```

If you visit the address `http://[your Pi's IP address]`, you'll see a page that lists all of the files in `/srv/http`. However, as the directory is currently empty, there will be nothing listed there.

10 CGI Hello World

Change directory into `/srv/http` using the command `cd /srv/http` and create a new file called `index.cgi` using the command `touch index.cgi`. Mark it as executable using `chmod +x index.cgi`. Once you've done that, open it in your favourite editor. The first line tells Apache it needs to use Python to interpret the script.

The first line starting with 'print' tells Apache to expect HTML content. The print line following that prints a blank line, and then the lines after that print a standard HTML hello world page. Notice how we use the three quotation marks to print multiple lines conveniently.

Refresh the empty index page you saw before and you'll see 'Hello World!'.

```
#!/usr/bin/env python2
```

```
# Tell the web server to expect HTML
content
print "Content-Type: text/html"
print
```

```
# Print a html hello world
print """
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<h1>Hello World!</h1>
</body>
</html>
"""
```

11 Creating a menu

So, the first thing we need to do is to create some links that let us choose what to do. We're going to use a couple of examples: one on running some command on the system and displaying the output; the other an example that could open and close a garage door.

You pass parameters to CGI scripts with the following syntax:

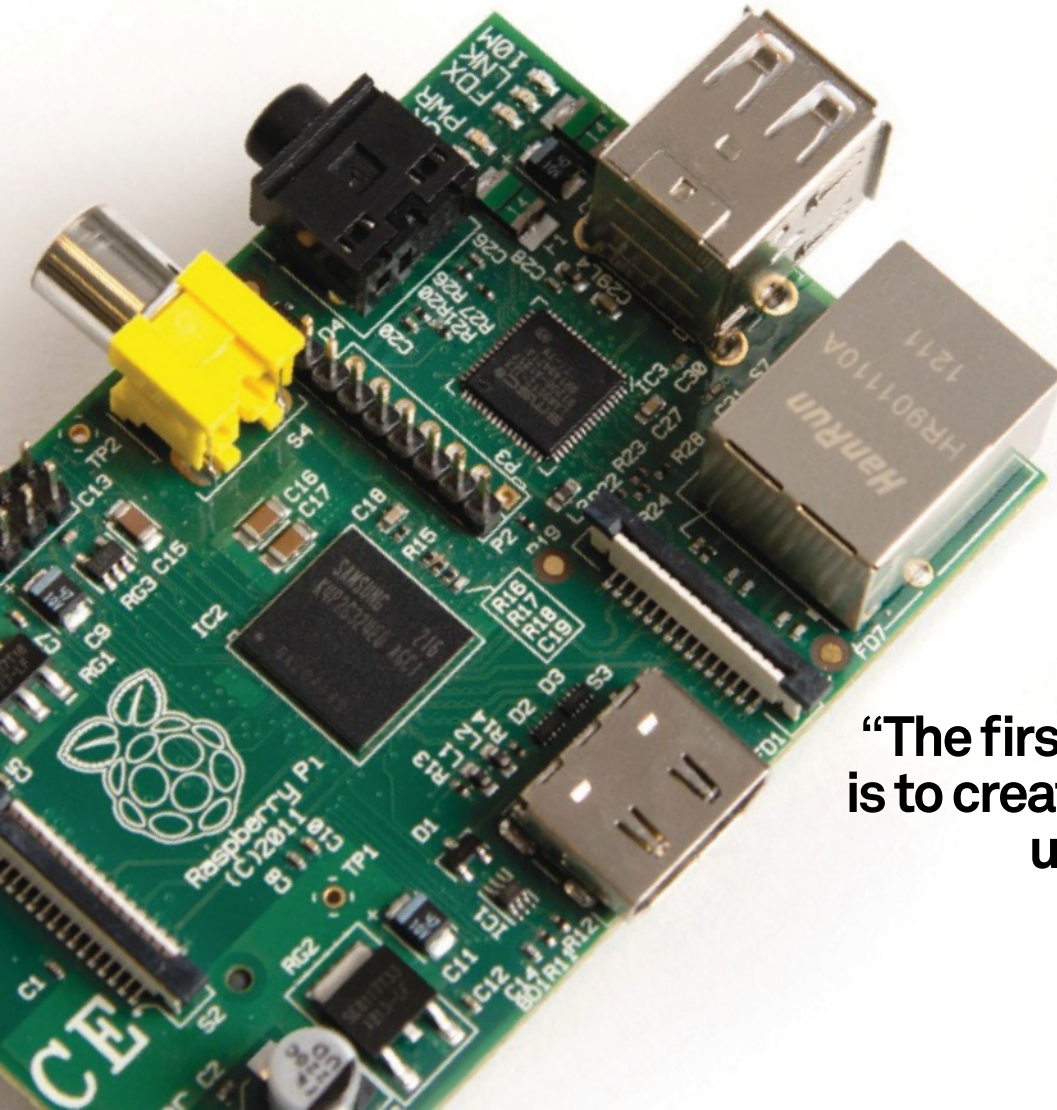
```
script_name.
cgi?var1=value1&var2=value2
```

Your author wrote some of the dispatching code and a menu that will cause some of that code to execute. Notice how he's used a function that prints the CGI header and then HTML page using the body and title that you specify, to avoid duplicating code unnecessarily.

At the moment, you will get a 500 error from the web server if you click any links, because nothing happens for any of the commands yet.

```
#!/usr/bin/env python2
```

“The first thing we need to do is to create some links that let us choose what to do”



```
# A CGI dispatcher written in Python
by Liam Fraser for a Linux User and
# Developer tutorial.
```

```
import cgi
```

```
def print_page(title, body):
    # Tell the web server to expect
    content
    print "Content-Type: text/html"
    print

    print """
<html>
<head>
<title>{0}</title>
</head>
<body>
<h1>{0}</h1>
{1}
</body>
</html>
""".format(title, body)
```

```
def print_menu():
    # Print a menu
    title = "Liam's Raspberry Pi"
    body = """
<p><a href="index.cgi?action=run_
command">Run a command</a></p>
<p><a href="index.cgi?action=garage_
control">Control the Garage</a></p>
"""
    print_page(title, body)
```

```
#
# Start of script
#
if __name__ == "__main__":
```

```
    # Get any parameters
    params = cgi.FieldStorage()
```

```
    # Variable to keep track of if
    we have a valid input or not
    valid = False
```

```
    # If we have a key called action
    in the params
    if params.getvalue("action"):
        action = params.
        getvalue("action")
```

```
        if action == "run_command":
            valid = True
```

```
        elif action == "garage_
```

```
control":
            valid = True

        if valid == False:
            print_menu()
```

12 A word about debugging

If you get a 500 error when you did not expect to get one, the first thing you probably want to check is that you have printed a header. Other than that, there may be a syntax error in your Python, but Apache isn't very useful when it comes to pointing that out for you. There are a couple of things you can do. The first is to exit your editor and run the code in the command line to see if Python exits with any syntax errors. You can do this by typing `./index.cgi`. You can also look in the Apache error logs for errors. You can keep an eye on the error log with the command `tail -f /var/log/httpd/error_log`. You can also print the last 50 (or any other number) lines using the command `tail -n 50 /var/log/httpd/error_log`.

13 Running a command

The following function displays a page which allows you to submit a command to run and then displays the output. If you look at the HTML form on the bottom, when the Submit button is clicked, the name and value will be sent to `index.cgi`. For example, if we wanted the output of `ps`, the request would look like: `index.cgi?action=run_command&cmd=ps`

Other than that, the code should be pretty self-explanatory. The only other thing you have to do to get it to work is add the import line:

```
import subprocess
```

...and call the function from the dispatcher part of the script, making sure you pass through the parameters:

```
        if action == "run_command":
            valid = True
            run_command(params)
```

Give it a try!

```
def run_command(params):
    # Deal with any run_command
    related tasks
```

```
        if params.getvalue("cmd"):
            # We have a command to run
            cmd = params.getvalue("cmd")
```

```
        # Subprocess.check_output
        needs an array of parameters
```



```
        # split by spaces into a
        list
        cmd_list = cmd.split()
```

```
        try:
            output = subprocess.
            check_output(cmd_list)
        except:
            output = "Error running
            command."
```

```
        title = "Output of {0}".
        format(cmd)
        body = "<pre>{0}</pre>".
        format(output)
```

```
        print_page(title, body)
```

```
    else:
        # Print the page where we
        submit the command
        title = "Run Command"
        body = """
<form action="index.cgi"
method="post">
Enter a command to run: <input
type="text" name="cmd">
<input type="hidden" name="action"
value="run_command">
<input type="submit" value="Submit">
</form>
"""
```

```
        print_page(title, body)
```

14 Permissions

Apache runs as the HTTP user; this means that it has very limited privileges. This is problematic, especially because you're probably going to need root permissions for any kind of project involving home automation to access the various data interfaces on the Raspberry Pi. To solve this problem, we're going to use sudo. Change directory to /etc/sudoers.d and create a new file called http. Then open it in your favourite editor.

An obvious thing to test that won't work is the reboot command. First, you'll need to find out where the reboot script is located, using `whereis reboot`. To allow the HTTP user to execute this with sudo, add a line like this:

```
http ALL=(ALL) NOPASSWD: /usr/sbin/ip, /sbin/reboot
```

As shown, you separate multiple commands the user can run with commas. If you now type `sudo reboot` at the run-a-command screen, the Pi will restart.

15 Garage control example

Your author has written an example that you could use if you were controlling a garage door. The example is pretty straightforward and very similar to the one that runs a command above. As before, you have to call the function near the bottom of the script, by adding the line `garage_control(params)`

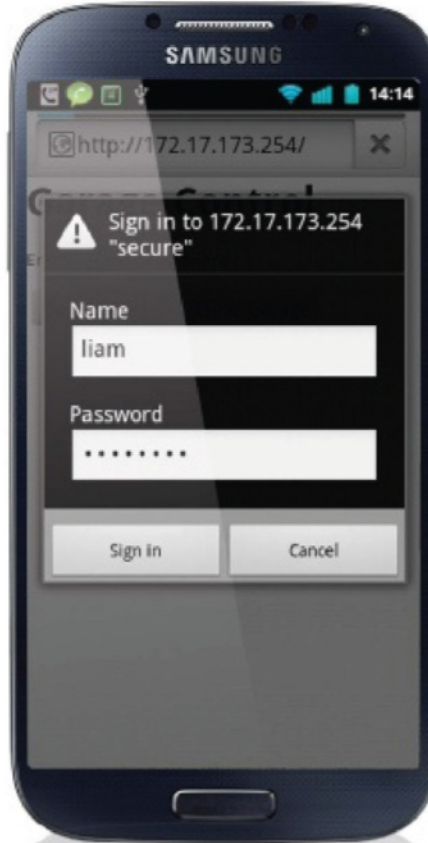
```
def garage_control(params):
    # Define output up here so it's
    # in the scope of the entire function
    output = ""
```

```
    if params.getvalue("garage_
    action"):
        action = params.
        getvalue("garage_action")
```

```
        if action == "Open Garage":
            cmd = "sudo /script/to/
            open_garage"
```

```
        elif action == "Close
        Garage":
            cmd = "sudo /script/to/
            close_garage"
```

```
        if cmd:
```



```
            # Execute the command
            cmd_list = cmd.split()

            try:
                output = subprocess.
                check_output(cmd_list)
            except:
                output = "Error
                running command."

            title = "Garage Control"
            # Create the body including the
            # output of a command if we ran
            # one
            body = ""
            <form action="index.cgi"
            method="post">
            <input type="hidden" name="action"
```

```
value="garage_control">
<pre>{0}</pre>
<input type="submit" name=garage_
action value="Open Garage">
<input type="submit" name=garage_
action value="Close Garage">
</form>
""" .format(output)
```

```
print_page(title, body)
```

16 Authentication

If you are planning to expose anything to the internet, you're going to want some authentication on it. You may also want HTTPS, where the HTTP connection is encrypted, but configuring that is outside the scope of this article.

The first thing we need to do is create an authentication file with the `htpasswd` tool:

```
[root@remotepi httpd]# htdigest -c /
etc/httpd/auth secure liam
Adding password for liam in realm
secure.
```

```
New password:
```

```
Re-type new password:
```

For all subsequent users, remove the `-c` flag. Now we need to edit the Apache config file again, and add the following lines just after the `AddHandler` line that we added to the `<Directory /srv/http>` section before:

```
# Authentication
AuthType Digest
AuthName "secure"
AuthDigestDomain /
AuthDigestProvider file
AuthUserFile /etc/httpd/auth
Require valid-user
```

Once you've done this, restart Apache with `systemctl restart httpd` and try to visit the site. You'll be prompted for a username and password and not let in otherwise.

17 Setting up Dynamic DNS

Head over to www.no-ip.com/personal/ and sign up for the No-IP Free option. Once you have done that, don't bother downloading No-IP's client because we've already installed it. Go to your email inbox and follow the activation link that was just sent to you by No-IP. You can now sign into your account. Once you have logged in, select the 'Add a host' option. Choose a hostname and a domain to be part of from the drop-down list. Leave the Host Type as 'DNS Host' and then click the Create Host button. Your

“An example that you could use if you were controlling a garage door”

author used the hostname liam-ludtest with the domain no-ip.org, so we'd access that using liam-ludtest.no-ip.org.

18 Configuring No-IP

Run the command `noip2 -C -Y` to be taken through interactive configuration of the No-IP client. We've left the update interval to the default of 30 minutes, meaning that the client will check every 30 minutes for an IP address change.

Once you have finished, start the daemon with the command `systemctl enable noip2` followed by `systemctl start noip2`.

After a minute or two, your IP address will be accessible via your No-IP hostname. However, it's likely that trying it from inside your own house will simply take you to your router's homepage.

19 NAT port forwarding

It is likely that there are multiple devices behind your router that all use the same external IP address. This is due to the shortage of IPv4 addresses, and because it is more secure to segregate the internet from your internal home network. NAT (network address translation) forwards a port from the router's external IP address to a computer on the LAN. In this case, we'll want to forward any traffic for TCP port 22 that comes to your router's external IP address to the IP address of your Raspberry Pi. TCP port 22 is the port used for SSH. You'll also want to forward TCP port 80 for HTTP traffic.

The configuration of port forwarding really depends on the router that you are using, so you may have to look it up. The chances are that it will be hidden away in the 'Advanced' section of your wireless router. You should be able to access your router by typing your No-IP hostname into your web browser. If not, it should be at the address of your default gateway that we used earlier on.

On your author's router, he had to go to Advanced>NAT>Port Mapping and add a mapping that looks like the one in the screen above (but with a different IP address) for SSH. He then had to add another for HTTP.

20 Testing it out

That should be it! You'll have to test the No-IP hostname from outside of your network to verify that it's working. Obviously, this is a very basic example of writing a web application. Hopefully you can put it to good use!

Hostname Information

Hostname: liam-ludtest no-ip.org

Host Type: DNS Host (A) DNS Host (RoundRobin) DNS Alias (CNAME)

Port 80 Redirect Web Redirect AAAA (IPv6)

IP Address: 78.145.93.187

Assign to Group: -No Group- [Configure Groups](#)

Enable Wildcard: Wildcards are a Plus / Enhanced feature. [Upgrade Now!](#)

Interface	Protocol	Remote Host	External Port	Internal Port	Internal Host	Mapping Name	Enable	Remove
nas_0_38	TCP		22	22	192.168.1.191	SecureShellServer(SSH)	Enable	<input type="checkbox"/>

Settings

Type: Customization Application Please Choose...

Interface: nas_0_38 Protocol: TCP

External port: 22 Internal port: 22

Internal host: 192.168.1.191 Remote host:

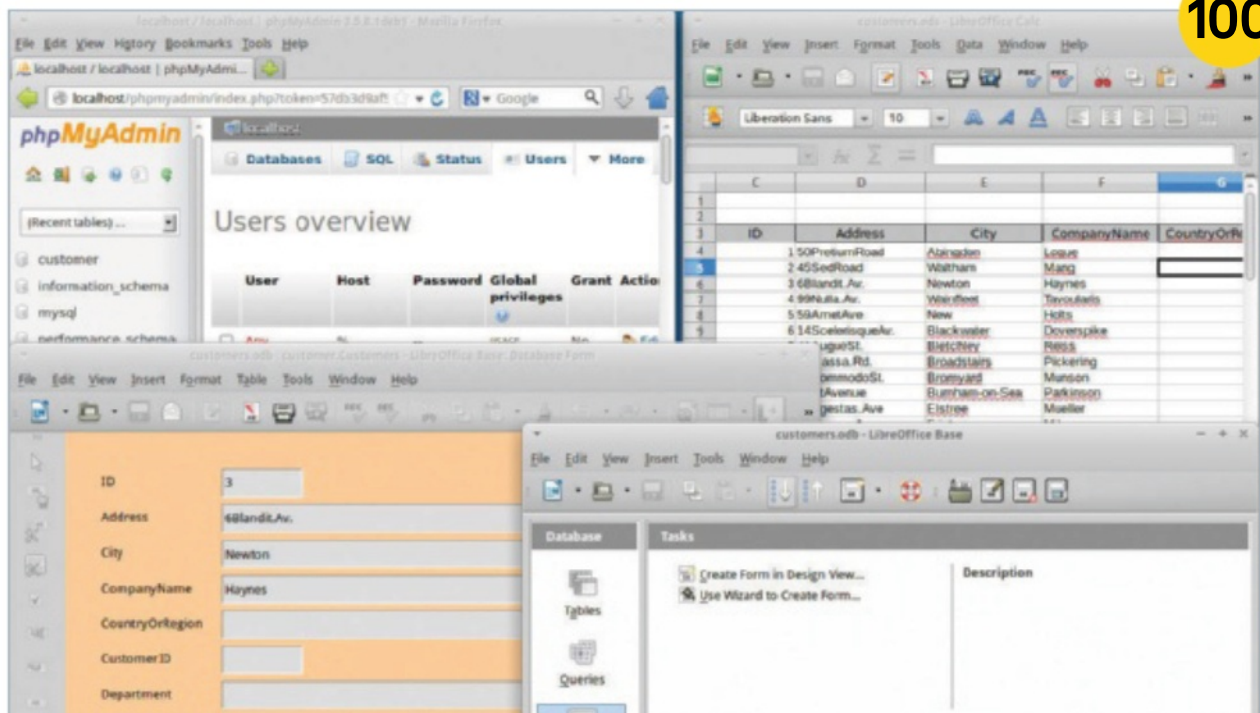
Mapping name: SecureShellServer(SSH)

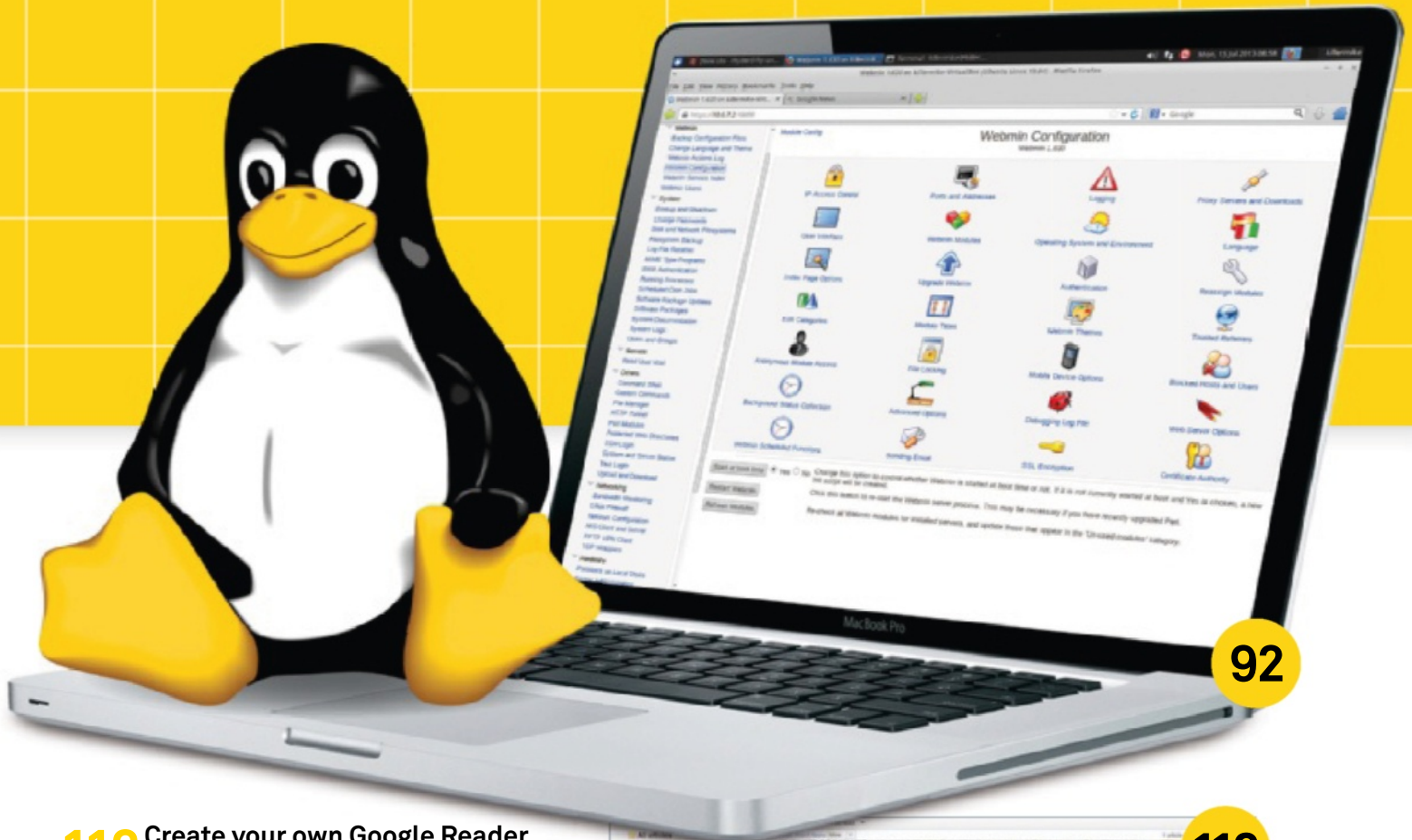


Tricks

Unlock the true power of open source

- 80** **Dual boot with Windows 8**
Conquer Secure Boot and get yourself dual-booting Windows 8 and Linux
- 88** **Network wirelessly with wicd**
Get your wireless network up and running
- 92** **Manage your system with Webmin**
A great GUI front-end for system configuration
- 96** **Synchronise your files with Unison**
Use this command-line tool to sync files between computers
- 100** **Make a small business database with LibreOffice**
Create an easy-to-use, form-based database
- 104** **Create and save data with a MongoDB database**
Forget about joins and SQL and try NoSQL databases
- 108** **Maintain and manage all of your machines with Puppet**
Keep them in a consistent and workable state





92

112 Create your own Google Reader
Use Tiny Tiny RSS to create a browser-accessible RSS feed

116 Visualise directory structures with Graphviz
Make large directory structures practical

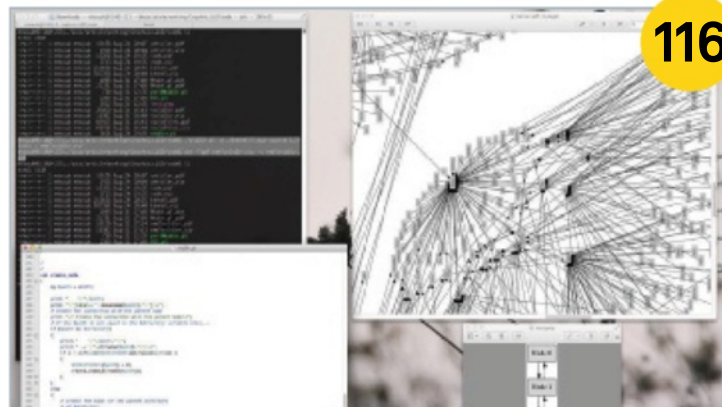
120 Create professional presentations with LaTeX
Customise LaTeX's beamer class to get a professional look

124 Make the most of open-source genealogy with Gramps
A fun hobby for many people around the globe

128 Take pictures and video with the Raspberry Pi Camera
Get a Pi Camera set up on your Raspberry Pi

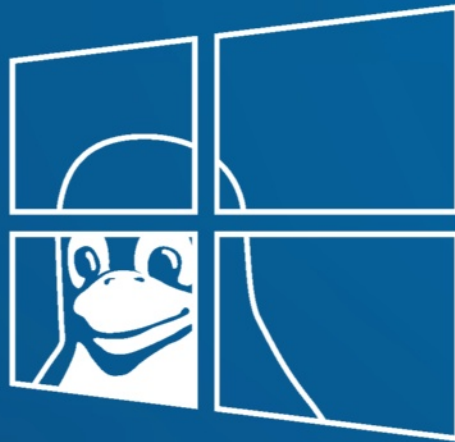


112



116

“The Raspberry Pi camera can be a lot more versatile than a standard webcam”



Dual Boot with Windows 8

Conquer Secure Boot and get yourself dual-booting Windows 8 and Linux on both your UEFI and BIOS-powered system...

The release of Windows 8 in 2012 brought to a head the long-running controversy over the divisive and restrictive Secure Boot function, and how it would hinder or outright block users from installing Linux distros onto the majority of new hardware. Since then, the Linux community has been adjusting to these changes, with some distros adding Secure Boot compatibility, and the Linux Foundation itself creating its own solution.

Dual booting is incredibly useful in today's computing space, with the ability to develop cross-platform to real hardware, or use OS-specific software without having to use multiple

machines. A lot of distros are also dual-boot minded, with options during installation to aid in dual booting with other machines.

Secure Boot shouldn't prevent you from using dual booting on your system, either. There are a number of methods to either fix, shim or avoid having to deal with Microsoft's controversial security feature at all. In this feature, we're going to be covering dual booting by adding Linux, Windows, or starting entirely from scratch. Ideally, installing both operating systems at the same time will result in the best outcome, but that might not be possible for every setup.

Understanding and disabling Secure Boot

Secure Boot is an option present in UEFI, the firmware interface that is in the process of replacing the venerable BIOS. This is used to prevent operating system loaders or drivers that are not signed with a specific key from running on the system.

While this is optional, the main concern from the Linux community was Microsoft announcing that all new Windows 8 PCs sold would have Secure Boot turned on by default. This is mandatory for all hardware manufacturers that ship Windows 8 with their systems, which can be problematic to Linux users who are buying new, preconfigured hardware due to the lack of OS-less computers on the market.

This wasn't the end of the story, though, as the Linux Foundation announced that it

was securing a signed key, and a method of using it, for the entire Linux community to use. There are also a handful of Linux distributions that now feature support for Secure Boot, such as Ubuntu and Fedora. These distros make it easy to run them with Secure Boot, and this filters down to distros based on them, such as Linux Mint.

Disabling Secure Boot on systems is fairly easy. Entering the UEFI setup, you should be able to find the option to disable it, which we'd recommend if you're creating your dual-boot system from scratch. Check the manual for your make of motherboard if you can't find it, though, as it may not be conveniently labelled as Secure Boot. For systems that are already using Secure Boot, you cannot disable it unless you wish to wipe the system and start again.

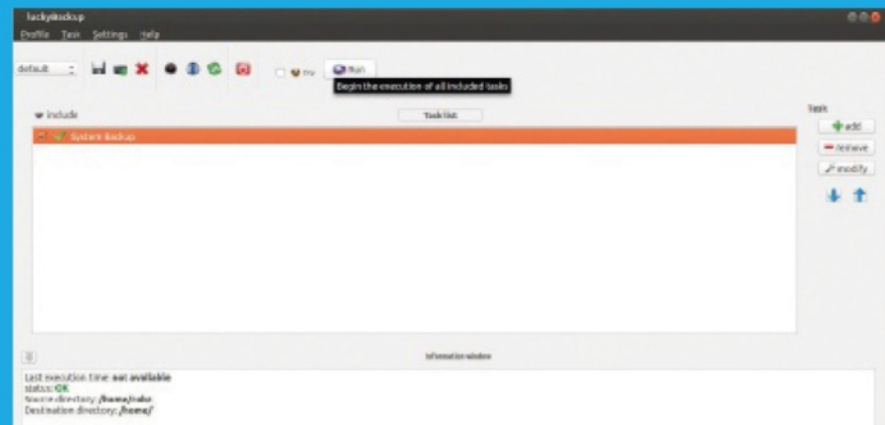
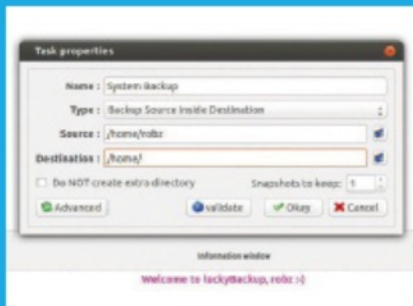


■ The Unified Extensible Firmware Interface is an extension of the original EFI

Back up now!

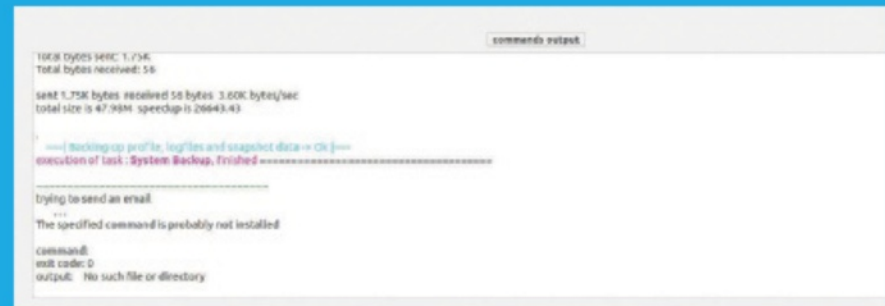
If you're using the methods that involve modifying existing installations, we highly recommend to get your systems backed up before proceeding any further. While the steps we're taking will be safe, problems and accidents happen. The simplest method would be to back up your important files to online or external storage. However, if you need to do a more rigorous backup, we suggest using the cross-platform, open source luckyBackup. Make sure that any backup files you create are then saved externally, just in case the worst happens.

01 Create task Click 'add' on the right column to set up a new backup task. Choose what you wish to back up, and a destination (preferably an external source). Make sure that the type is 'Backup Source inside Destination', then click the Okay button.



02 Run task From here, check the task in the list and click the Run button. If you wish to use this more in the future, you can set up a cron job via the interface to schedule backups or check past backups if you're running it for a second time.

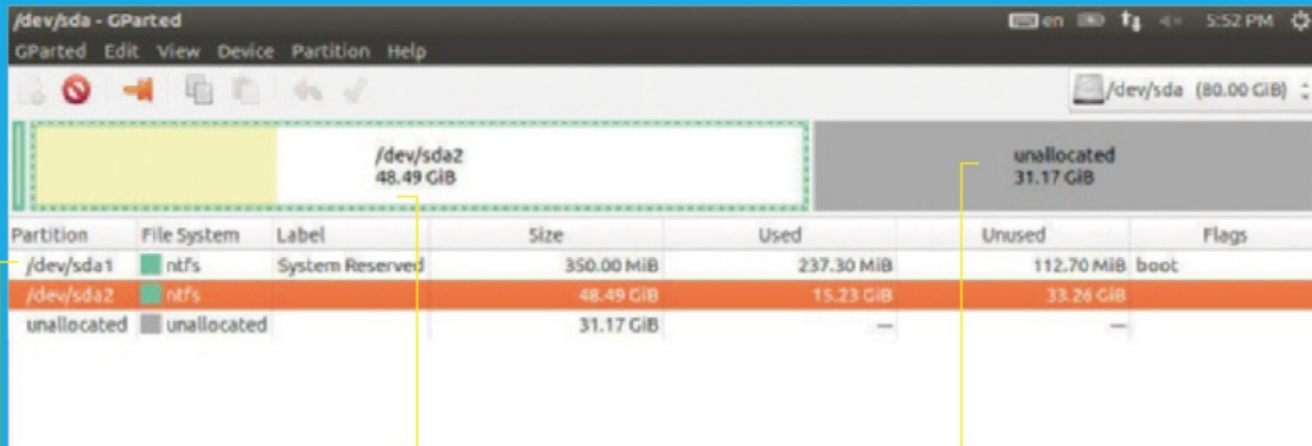
03 Complete backup LuckyBackup will run through the system and make copies of the files you're trying to back up. Look out for any error messages and make sure to address them. If you didn't make the backup destination external, then move it externally now.



Windows partitioning

Before installing a Linux distro to your hard drive, you'll need to make sure you have space for it. Windows has its own partitioning utility; however, we prefer to use GParted. It's included on most live Linux distros, and some recovery distros, and is easy enough to use. You'll

need to shrink the primary Windows partition and free up at least 20GB of space – that will give you enough room to store extra data other than just the distro and packages. Defragment Windows and back up before doing so, though, or you might lose some files.



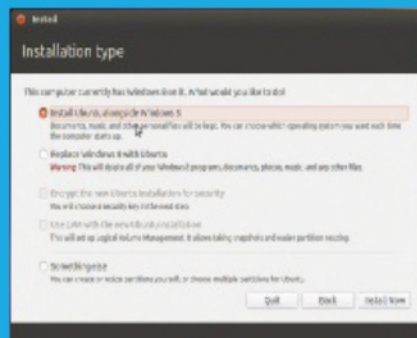
By default, Windows will all be installed to one primary partition, which is here. Resize this

This is the main boot partition used by Windows – do not alter this

Here's the remaining space we'll use for the Linux distribution

Installing alongside

Some modern distros offer you the option to install alongside – this should automatically use the remaining space on the hard drive for the necessary partitions. However, not every single distribution uses this, and you may want to leave some for another distro, or an NTFS-formatted home partition so you can share your files with the Windows side. At the very least, though, it's a quick option that is fairly safe – much safer than using it before you've edited your Windows partitions. Ubuntu, specifically, will attempt to resize it for you, but not to your liking.

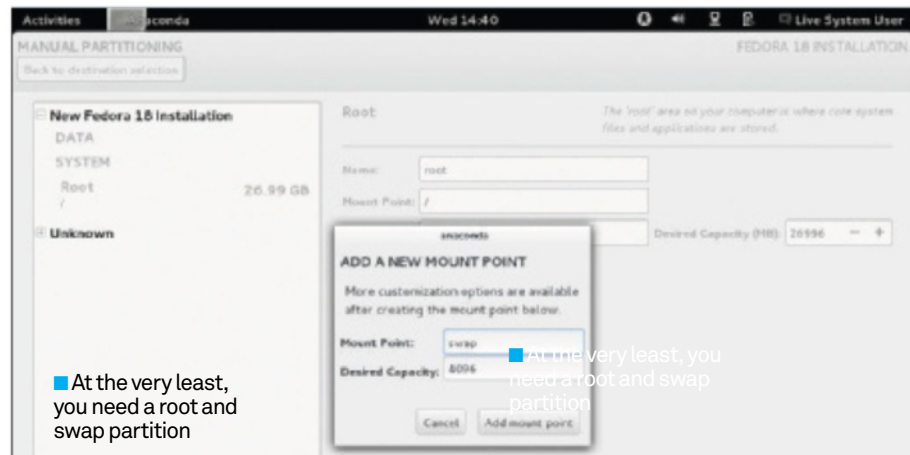


■ The quick route is not always the best

Installing Linux

For all situations, you'll need to make sure you set up the rest of your partitions for installation in the correct way. At the very least, you'll need to add two partitions – a root partition and a swap partition. Swap partitions can go right at the end of the hard drive space and need to be the size of available RAM in your system. You can either fill up the rest of the space with your root partition, create a home partition as well, or leave space for a third distro. Whichever, make sure to set the root flag on the correct partition.

For Secure Boot-enabled systems, the Linux distributions that support it will either boot without an issue, or require you to go back into the UEFI setup and temporarily switch off Secure Boot and revert to BIOS Legacy mode. Ubuntu, Fedora and openSUSE will recognise the EFI boot partition and automatically install the relevant files to that partition. Once it's all installed, turn BIOS Legacy mode off and switch Secure Boot back on if the system complains about it.



■ At the very least, you need a root and swap partition

■ At the very least, you need a root and swap partition

Booting into Linux

On a Secure Boot system, your computer may boot straight into Windows without showing off GRUB. If this is the case, open boot selection on startup (usually indicated on the first screen as F9 or F12 etc) and select your distro from the list. If you're still having problems, follow the 'Repairing boot' steps on this page to fix any common issues. If the recommended repair doesn't work, though,

go to the advanced options and to the GRUB location tab. Make sure that the /boot/efi partition is selected and try again, and follow any instructions it provides for removing and then reinstalling GRUB.

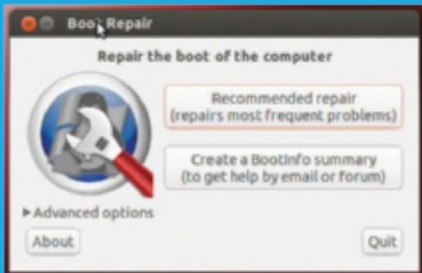
Currently, the shim and Linux Foundation's solution need to be implemented while building the ISO. Companies and dev teams are implementing these solutions in their own

way and at their own pace right now. However, if you want to build an ISO from sources for a distribution that currently doesn't have Secure Boot compatibility, refer to the 'More information' section of this page.

You can get a Linux entry on the Windows boot menu by editing the BCD, and we'll also cover editing your GRUB menus at the end of the tutorial.

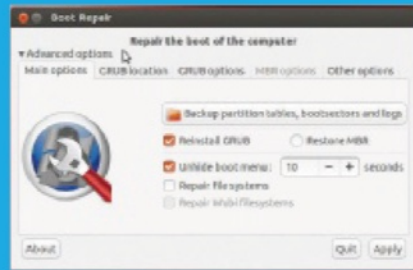
Repairing boot

A good portion of boot problems can be fixed automatically with boot-repair on the 64-bit version of Linux Secure Remix, which is well worth a look if you're having issues. It supports multiple bootloaders, BIOS, UEFI, Secure Boot and more. You can download it here: bit.ly/Z9gaqC



01 LinuxSecure

Live-boot into Linux Secure Remix and find boot-repair in the Unity toolbar. In most cases, you can just use the Recommended Repair option, which will automatically try to fix any issues with the system. This will work fairly well on most systems.

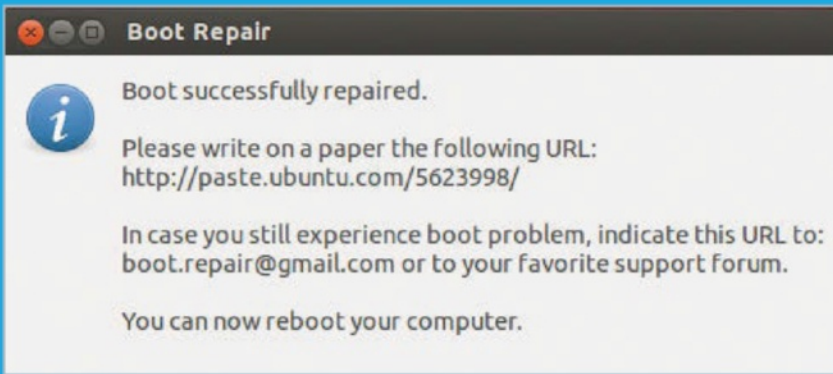


02 Advanced repair

You can select some advanced options to customise some of the ways it will try to repair the boot menu. This includes restoring the MBR, changing some timeouts and the location GRUB will be stored to. This can be avoided unless the automatic repair does not work.

03 Boot repair

The tool will run through the various tasks it's performing. Once it's finished, it will give you a URL to an output of the boot-repair just in case it doesn't work. You can use this to diagnose any further problems you have with the bootloader.



More information

For more information regarding other Linux distributions and Secure Boot, you'll need to check out their official websites for plans.

Alternatively, if you want to create your own compatible ISO, look up Matthew Garret's shim here:

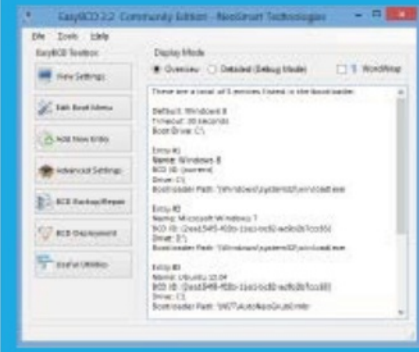
bit.ly/o174rm

Or the Linux Foundation's efitools here:

bit.ly/ZAnVal

Using EasyBCD

EasyBCD is a utility for Windows that allows you to edit the Windows bootloader, adding not only other Windows installs you have on the system, but also any Linux distros. This may in some ways be the easiest method to boot into Linux on a Secure Boot-enabled system, chainloading straight into GRUB. To use it, go to Add New Entry, then click the Linux/BSD tab. Make sure to select the right kind of GRUB, most likely GRUB 2, and then click Add Entry to finish.

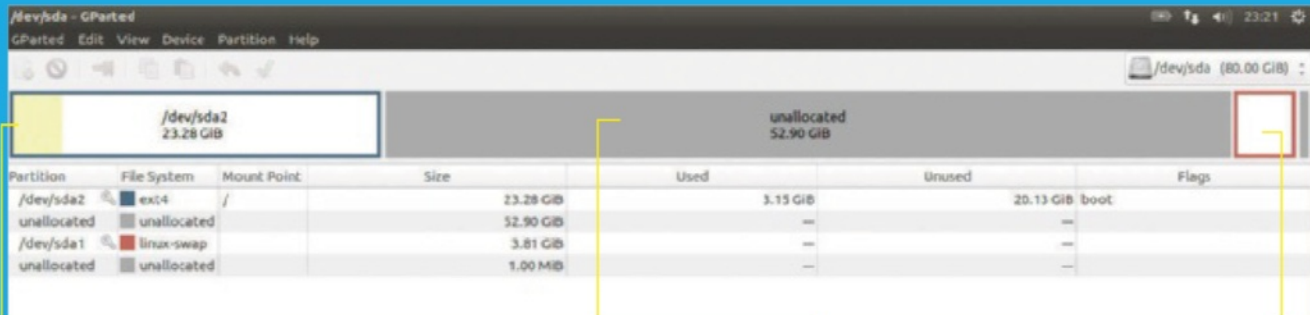


■ A Windows solution to boot menus

Linux partitioning

Much like how we partitioned Windows, you'll also need to make room on a Linux system to install Windows 8. Microsoft recommends a minimum of 20GB for a Windows 8 installation; however, we like to give it a bit more, as it fills up a lot quicker than Linux.

Unless you made your own specific partitions, you'll most likely have a root partition and a swap partition; or root, home and swap. Reduce the size of the home partition if you have it, or root if you don't, but leave the swap partition alone.



Our root partition on this system also contains our home folder, so we've reduced its size

The space we've made for Windows is more than enough

This is our swap partition, which we've kept at the end

Installing Windows



01 Boot from disc

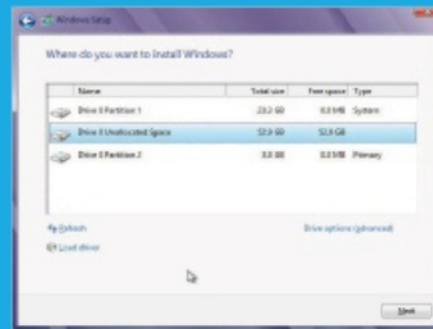
Insert the Windows installation medium, then select 'boot from disc' in the boot menu. It will take a moment to load up the installer. Select the language you wish to use for Windows, then click Next.

02 Install

Windows discs offer an alternative to live booting, with some repair options available that are either automatic or require



use of the command line. You can also come back to here to import old Windows files and settings. For now though, click install.



03 Partition

After accepting the licence agreement, you'll need to select how you want to install. Select the Advanced

option and the installer will search available storage. The Linux partitions will turn up as 'unknown', but you can select the remaining space to install it.

04 Wait

Windows will install all the files it needs, restarting once or twice in the process. Leave it to do its thing, including ignoring a prompt to 'Use any key to boot from DVD'. Once this is done, it will drop you into the initial startup process.

05 Personalise

It's up to you how you personalise your version of Windows, and whether or not you want to tie any online accounts to it. For the moment you'll only be able to boot into Windows, as the installation will have overwritten GRUB from the MBR.



Recovering GRUB

Installing Windows will write over the MBR with the Windows bootloader, removing GRUB's priority in the process. While we discussed previously how you can edit the Windows bootloader to chainload into your distro's GRUB, you can also reinstall GRUB and use that to specifically chainload into Windows. The method is different depending on what version of GRUB your distro is using, although a majority of recent distros will most likely be using GRUB 2. Recovering GRUB 2 is quite easy, and allows you to simply add Windows and other operating systems to the boot menu.

First of all, boot into a live CD of Fedora, Ubuntu or another distro that supports GRUB 2. You may need to install the necessary GRUB files in the live environment, either grub-efi or grub-pc depending on how your system boots. Once that is done, mount the boot partition of Linux (this will be the root partition if you haven't specified for the boot partition to be different) and then enter the following in a terminal:

```
$ grub-install --root-directory=/
[mount point] /dev/sda
```

We're assuming you want to install it to /dev/sda, the first (or only) hard drive in your system. Once that's done, you should be able to reboot into your normal distro. GRUB 2 has a script that looks for other operating systems on your computer, which will allow you to add Windows to the boot menu. In a terminal, run:

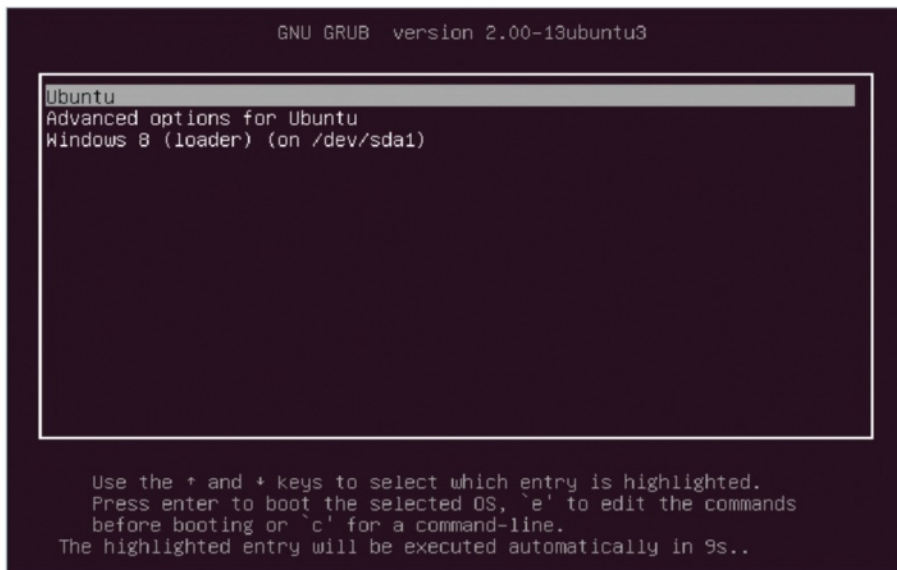
```
$ update-grub
```

Windows should now be on the boot menu along with your other distro.

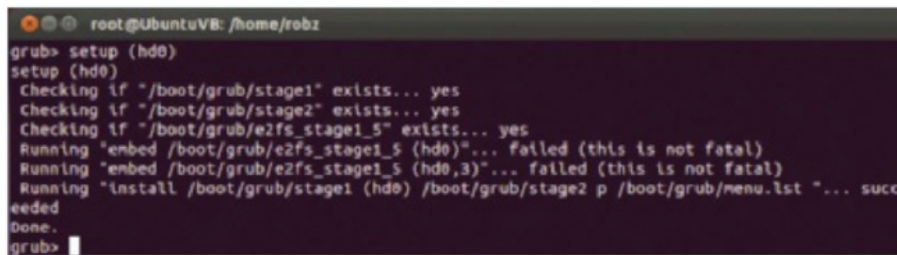
Recovering GRUB Legacy is a little different, and a bit more involved. Boot into a live CD, like before, and make sure the grub package is installed. Open a terminal and type grub to get into the GRUB command line so you can enter:

```
$ find /boot/grub/stage1
```

This will list the available GRUB stage 1s on the



The recovered GRUB will let you boot into both distros



GRUB Legacy requires you to know where your GRUB files are kept

Understanding file systems

You can't choose what file system Windows will install as – it will always do so as NTFS. Natively, Windows cannot see ext file systems, or indeed swap; however, Linux can easily see NTFS and any other Microsoft file systems like FAT 32. If you want to have cross-saving between both operating systems, you'll need to either save files to the Windows partition from Linux, or create an NTFS home partition. NTFS fragments much more than ext systems do, though, so your Linux system will not be as fast if it has to use it.

hard drive; in the case of installing Windows after Linux, it will likely be (hd0,1), depending on your partitions. Next, set that as the root with:

```
$ root (hd0,1)
Write the information to the MBR using:
$ setup hd0
=...and then finish with:
$ quit
```

Like before, this should restore GRUB and allow you to boot back into Linux. GRUB Legacy does not have the same OS prober as GRUB 2 does, though, so you'll either have to add Windows manually or use a separate tool.

Rescue your system

The ways we've described here can be used by pretty much anyone with any distro. However, there are recovery and rescue distros that include not only the tools we used, but graphical and automatic tools to make the process easier. We found SystemRescueCD to be the best recovery disc (see page 183 for a full review of the product). However, Parted Magic also includes some great GRUB tools, and there are specific GRUB rescue distros like Linux Secure Remix (which we used earlier). You can get a bit more control with the method on this page, though.



Recovery distros can help you with GRUB

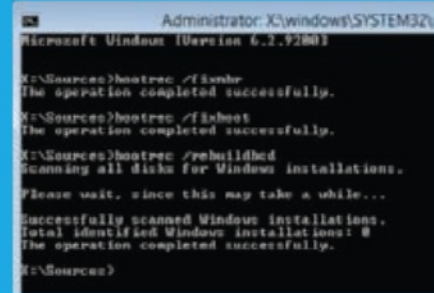
From scratch

The ideal way to get a dual-boot system is to start from a freshly formatted system. Because GRUB is generally better than the Windows bootloader for booting into multiple distros, we're going to install Windows first so that we don't have to do any recovery or rebuilding of GRUB after everything is done. You can create custom partitions while installing Windows as well, so you won't need to shrink anything afterwards. After installing Windows with these steps, refer back to installing Linux over Windows from earlier in the tutorial.

Using Windows boot recovery

If you wish to restore the Windows bootloader to the MBR and use EasyBCD for your dual-boot needs, you can use the recovery tools on the Windows installation medium to do this. Insert and boot from it like before, but instead of clicking Install, click on 'Repair your computer'. You'll then be given some extra options – select Troubleshoot, then Advanced options and finally Command Prompt. You'll then need to use three separate commands to reinstall the bootloader:

```
bootrec /fixmbr
bootrec /fixboot
bootrec /rebuildbcd
```



■ These commands only work from within the repair disc

Installing Windows 8



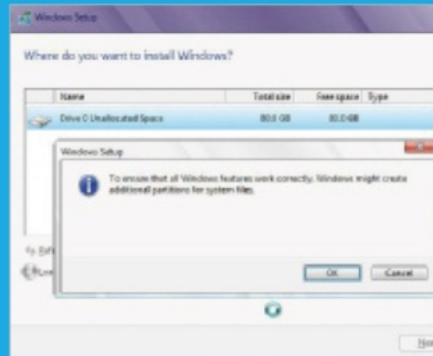
01 Install disc

Turn on your system with the Windows installation disc inserted, and make sure to boot from disc. You'll first be asked to choose your language, then you'll be given the option to install Windows. Click on Install to continue.

02 Hard drive

Windows will scan the hardware and display available storage and any partitions that might already exist. Select the hard drive

you wish to use, then click 'Drive options' to bring up the partition tool. Click New, then set the amount of space you want Windows to fill.



03 Partition size

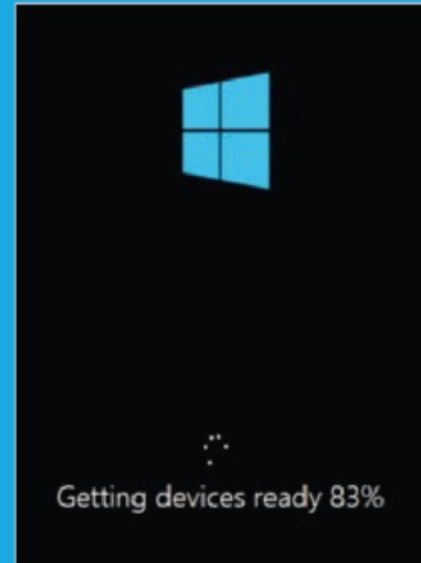
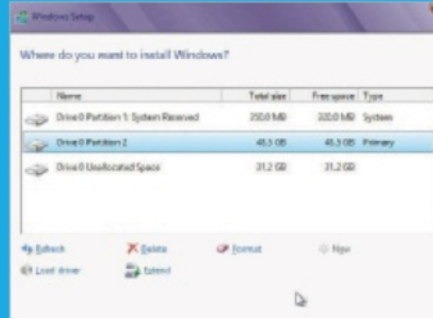
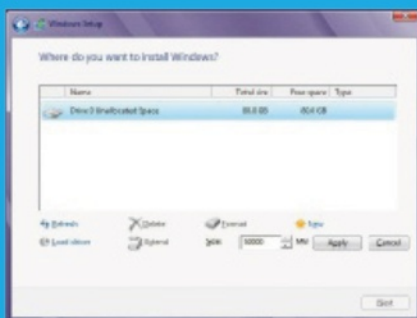
We opted to give Windows 50GB of space, which is more than enough for the basic install, with room for extra programs, folders and files. The installer will let you know that it may create other partitions, in this case a small boot partition.

04 Partition check

Make sure you're happy with the amount of space remaining. You can also add partitions or format extra hard drives at this point if you wish. Otherwise, click on Next to start the installation process with this partitioning.

05 Have patience

Windows will partition the hard drives and install all the necessary files, restarting once or twice in the process. Afterwards, you can continue with the setup process or go straight onto installing Linux – setup doesn't take too long, though.



Editing GRUB

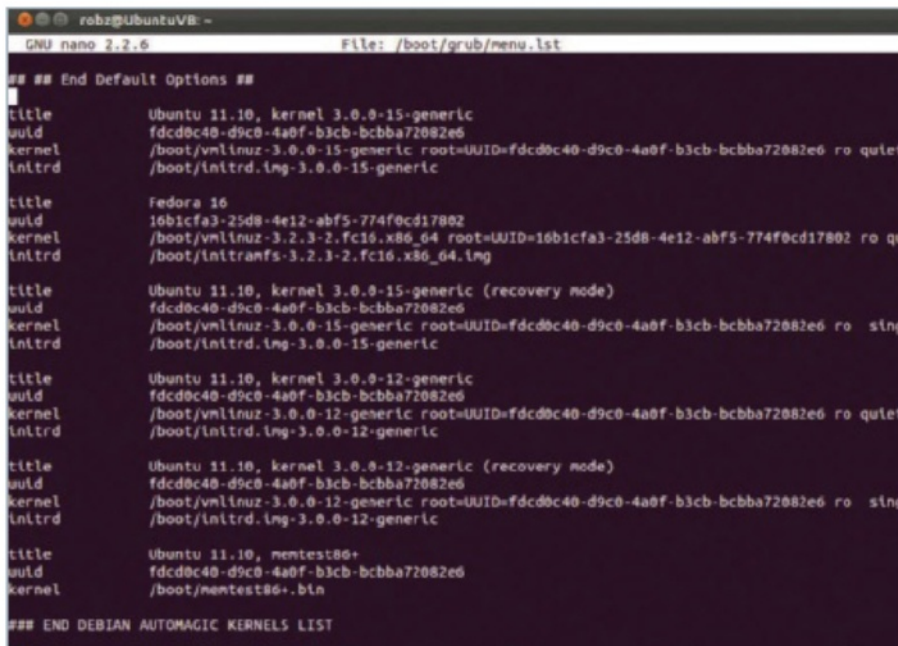
Now that you've got your new dual-booting system set up, it's time to start editing and moving around your boot menu to personalise it to your own needs. Again, we'll start by explaining how to do this in GRUB 2.

GRUB 2 organises boot entries by files in /etc/grub.d. However the files are numbered will be the way they're added to the boot menu when using grub-update in the terminal. The Windows OS entry will be done by OS prober by default, whereas the Linux entry is governed by the 10_linux file. Both are fairly generic and search the system for boot details, and make them up themselves. There are two ways, then, to change the boot order, or at least have Windows boot by default in GRUB 2.

The first, most obvious way is to change the number on OS prober. In our system it's



■ Changing the order of the files is easier in some ways



■ Editing GRUB Legacy just involves one file

30_os-prober, so changing it to 09_os-prober means it will come before the Linux entry in the boot menu. Otherwise, you can change the default entry in /etc/default/grub by editing the GRUB_DEFAULT=0 value.

As well as simply changing around the OS prober entry, you could also create your own Windows entry entirely. Create an appropriately numbered and named file and enter the following:

```
#!/bin/sh -e
echo "Adding Windows"
cat << EOF
menuentry "Windows 8" {
set root=(hdX,Y)
chainloader (hdX,Y)+1
}
EOF
```

GRUB Legacy works slightly differently, instead editing the menu directly with the entries. To find them, go to /boot/grub/menu.lst, and move around the entries to change the order of boot. To add Windows to GRUB, you'll need to do a menu entry like this:

```
title Windows
rootnoverify (hdX,Y)
chainloader (hdX,Y)+1
```

GRUB Legacy does partitions on a hard drive slightly differently from GRUB 2, so make sure you have the right combination. You can also change the default entry on the line that simply says 'default 0', changing the 0 to whichever entry you wish.

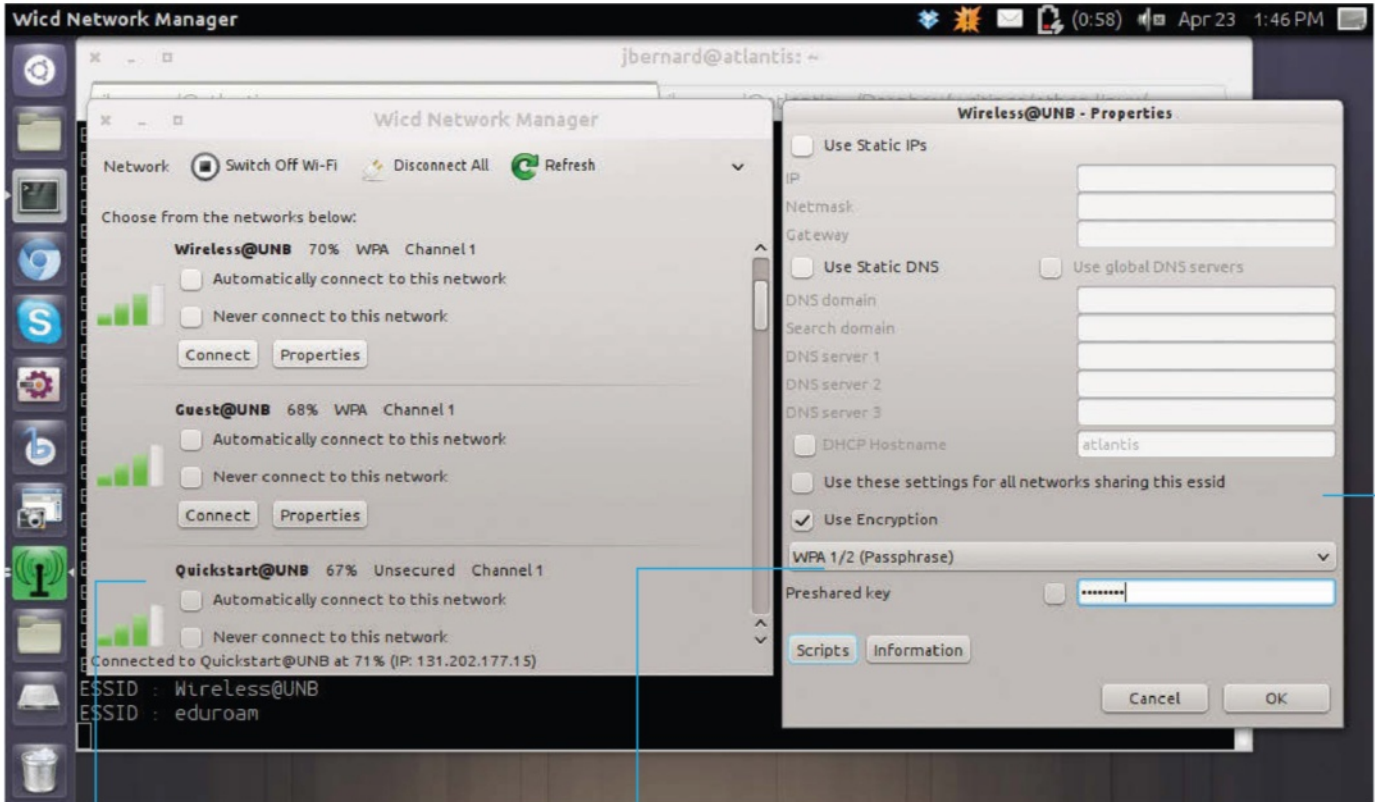
You can do much more to GRUB as well, including changing the colour of the text and background, using a background image, or even simple things like extending or reducing the timeout. It's a lot more customisable than the Windows bootloader, and supports all distros by default.

GRUB tools

Doing the above manually is the most universal way of editing your GRUB menus. However, there are specific graphical tools that will help you do it a little more easily. GRUB Customiser is one such tool, allowing you to change and add boot entries, move around the order, and set the default for the menu. Note that these tools don't work on every single system; however, some have their own specific tools, such as Mageia. The manual way, like before, offers you a lot more control.



■ GRUB Customiser is one of a few tools for customising your GRUB menu



This is the main list of wireless networks available to you. You can select the network to connect to and set its properties

Part of the properties you can set is the type of encryption being used, along with any special values that are needed, like passphrases

When you click on the Properties button, you get a window where you can set static IP properties

Network wirelessly with wicd

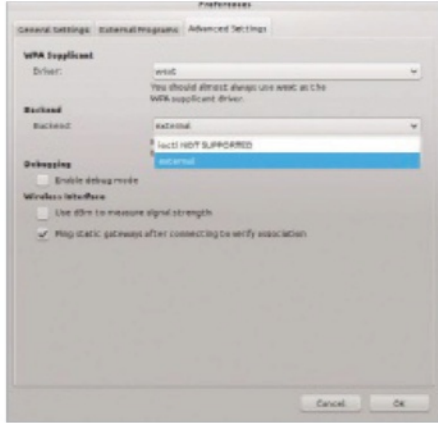
Wicd is a flexible alternative to NetworkManager, complete with interfaces for GTK, KDE, curses and the command line. Use it to get your wireless network up and running

Resources

Wicd: <https://launchpad.net/wicd> or wicd.sourceforge.net

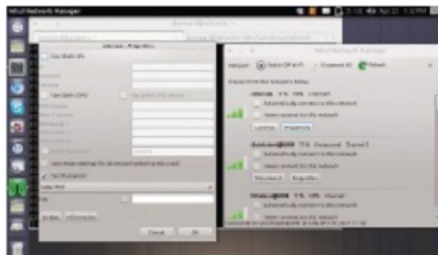
On most Linux systems, network management – both wired and wireless – is handled by a utility called NetworkManager. It is so ubiquitous that you may not even think about it. But, in Linux, there is always another choice. You can always do network management by manually configuring the appropriate configuration files. A better option is to use the utility wicd. Wicd provides interfaces using either GTK or KDE. This means you can use the

one appropriate for the graphic libraries for your desktop. There is also a text-based interface, which uses the curses library. You can even use wicd within your scripts or on the command line with the CLI interface. This tutorial will walk you through most of the interfaces, and how to use them to configure your machine's networking. This will include some issues, like using unusual setups of WPA security and adding functionality in the guise of network templates.



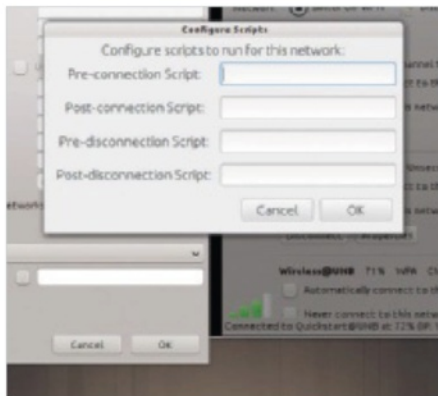
10 Back-end processing

Also in the Advanced Settings section is a selection for how to handle all of the back-end tasks to configure your network interfaces. The default (and most stable) is to use external programs, like iwconfig and dhclient. You can also choose to use IOCTL instead. It works faster, but is also more likely to fail.



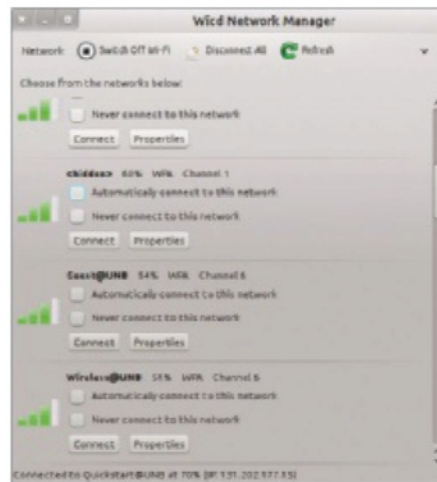
11 Interface properties

Each available network has its own set of properties. You can pull up the Properties window by clicking on the Properties button. Here you can set options if you are using a static IP address. If you are using encryption, you can select from the list of possible templates at the bottom of the window.



12 Scripts

At the bottom of the window, you also have the option of running scripts. There are options to set scripts to be run just before or just after connection, as well as just before or just after disconnection. This lets you customise connections to your needs.



13 Finding hidden networks

When you set up a Wi-Fi hotspot, you have the option of whether to broadcast the network name or to hide it from casual perusal. They'll still appear under wicd, labelled with the name <hidden>. This lets you find and connect to these hidden networks.

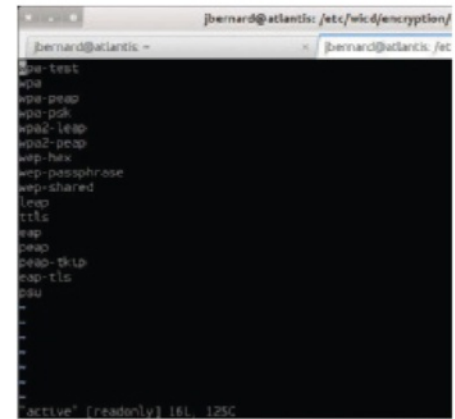


14 WPA templates

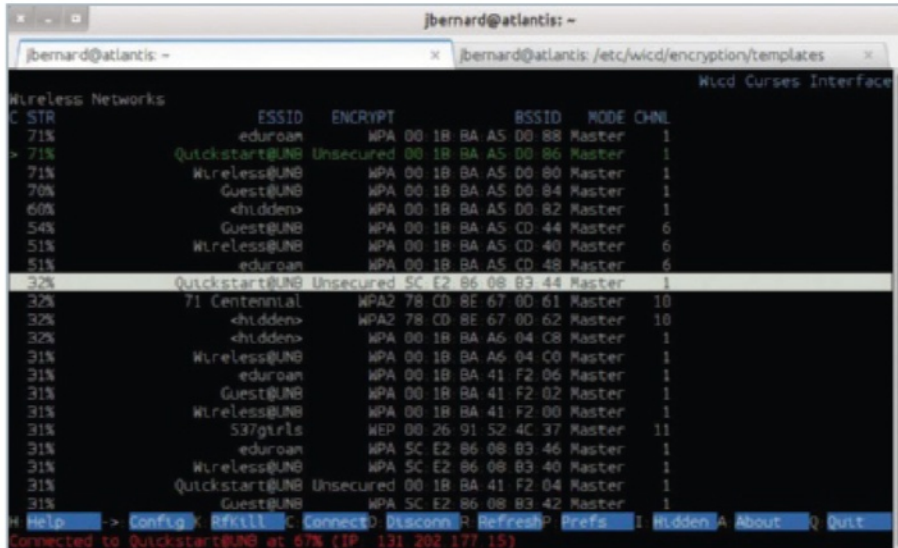
If the network you are using needs encryption, you can select the specifics from the list of available templates. These templates will change the remaining options in the Properties window and ask you for whatever values you need for that particular type of encryption.

15 Creating your own template

There are a surprising combination of options in encryption. So, wicd allows you to add templates for any combination of properties that wicd doesn't already support. The template files are stored in the directory /etc/wicd/encryption/templates. You should be able to find one that is already close to what you need. You can make a copy of this template and edit it to match the settings that you need. Once your new template is finished, you can add an extra entry in the file /etc/wicd/encryption/templates/active. It will then show up when you go to select the encryption template to use for your particular network.

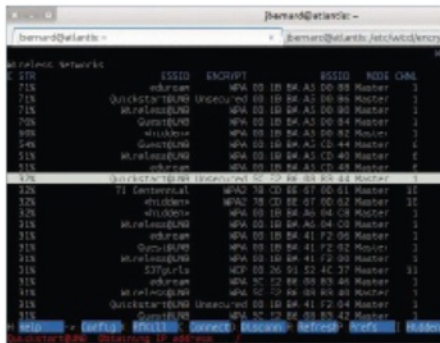


“Setting up Wi-Fi at the console is where wicd really shines”



16 Wicd-curses interface

There are several scenarios where you don't have a graphical interface but still need to set up wireless networking. Setting up Wi-Fi at the console is where wicd really shines. There is a text interface using the curses library that gives you all the same functionality that is available in the GTK version.

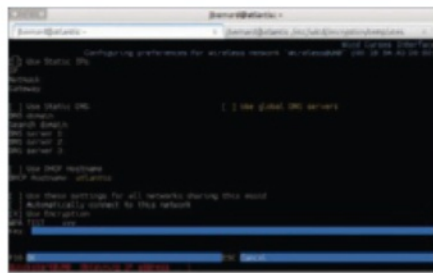


17 Connecting to a network

When the curses version starts up, it will show you the full list of available networks. You can use the arrow keys to move up and down the list to select the one you are interested in connecting to. When the correct one is selected, you can connect by pressing Enter.

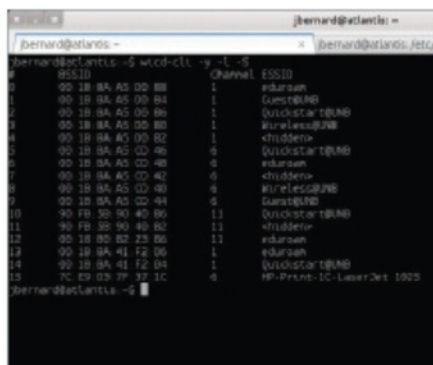
18 Changing preferences

If the network you are interested in uses encryption, you need to set the template. To access the Properties window, you need to select the network of interest and then press the right-arrow key. You can then set any static elements, and also set the appropriate template.



19 Wicd-cli Interface

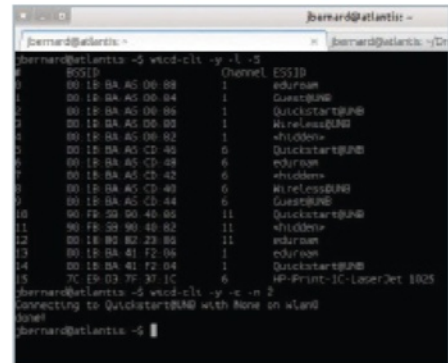
In some cases, you may not even have a terminal capable of curses display. For this situation, you have a command-line version of wicd that lets you set up and manage wireless networks with the most basic of text interfaces.



20 Scanning networks

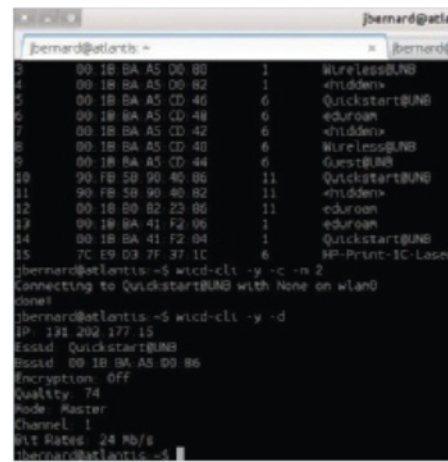
The first step is to scan for available networks. To look for them, you would run the command `wicd-cli -wireless -S`, or `wicd-cli -y -S`. This will do a scan, but not show

anything. If you want to see the results, you can either add `-l` to the above command, or subsequently run `wicd-cli -y -l`.



21 Connecting to a network

To connect to a given network, you would use `wicd-cli -y -c -n NETWORKID`. Disconnecting is done equivalently with `wicd-cli -x -y`. Setting options is a bit more involved, where you need to set individual properties with `wicd-cli -y -p PROPERTY -s VALUE`.



22 Listing active connections

You can see the details of your current network connection by using the command `wicd-cli -y -d`. This includes the name, type of encryption, quality and bitrate, among other items.

23 Where to now?

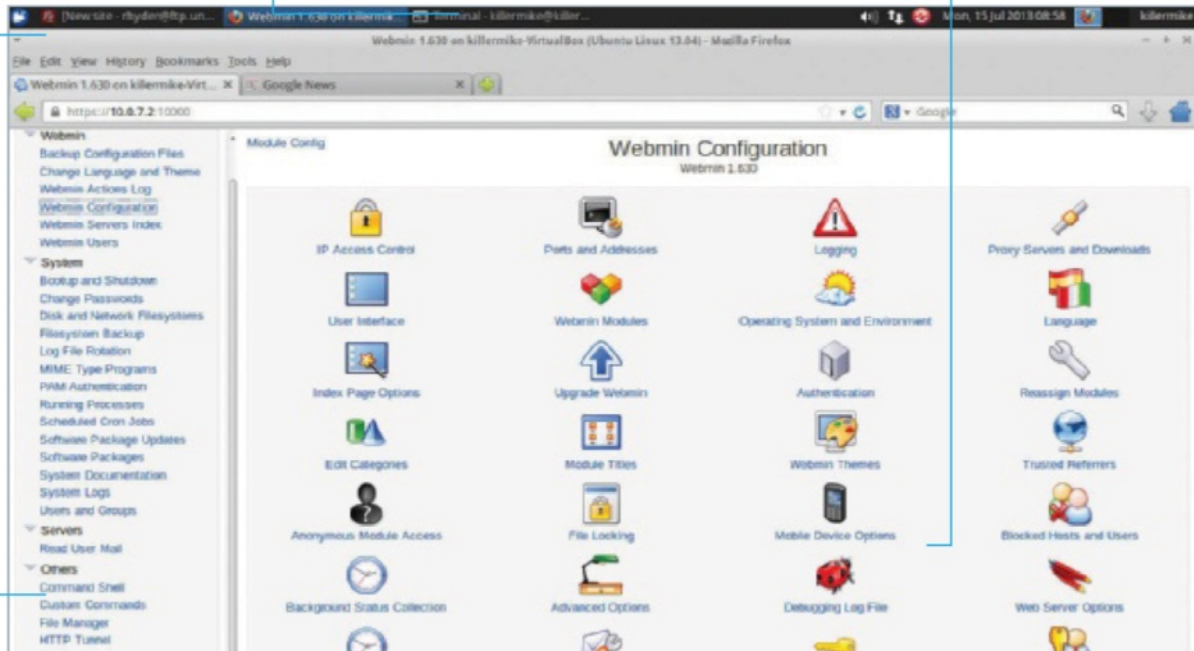
Now, with wicd, you should have all the tools required to easily configure wireless networking, no matter how basic a console you have. You can even build scripts that can handle the connection details at the proverbial touch of a button.

Access Webmin via a web browser, locally or remotely

You can access Webmin from pretty much any platform. This tutorial covers installation methods that should work for most distros

The main window within the web browser is where you interact with most of the modules

The facilities of Webmin are provided by a massive collection of modules. These modules are accessed via the sidebar



Manage your system with Webmin

As well as enabling you to administer a system remotely, Webmin is a great GUI front-end for system configuration

Webmin offers administration of a Linux system via a web interface. It is implemented as a set of Perl scripts that includes a small web server. Part of the appeal of Webmin is that it's extremely comprehensive as there are modules for most typical administration tasks. This includes core system areas such as management of printers and users, and package-specific tasks such as configuration of Apache and Squid. In addition, it includes some handy tools like basic file backup and transfer, and system resource monitoring.

Webmin isn't generally included in the package repositories for most Linux distributions, so setting it up entails downloading it from the website (www.webmin.com). This suits most

administrators since they tend to prefer carrying out updates of administrator tools when they are ready rather than as part of an automatic system upgrade. Webmin itself contains its own facilities for updating modules as updates become available.

So what can you do with Webmin? A typical example is configuring a system remotely, and we'll show you how to do that, but there's a lot more besides such mundane tasks on offer. Since it's so mind-bogglingly comprehensive, there is no reason why it can't be used as an all-encompassing system configuration GUI. In this context, it has the potential to offer a consistent organisation-wide configuration GUI that's backed up by a reassuring 15-year lineage.

Resources

Any Linux system

Webmin: www.webmin.com

01 Fetch the latest version

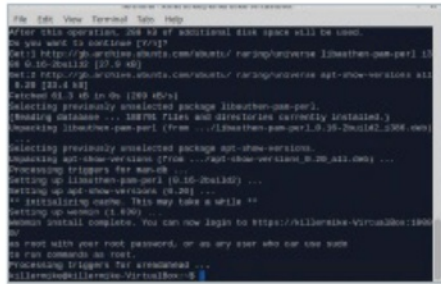
Head on over to www.webmin.com and proceed to the Download page to fetch the latest version of Webmin. Alternatively, download the latest version as a DEB file by typing `wget http://www.webmin.com/download/deb/webmin-current.deb`.

Or type `wget http://www.webmin.com/download/rpm/webmin-current.rpm` for RPM.



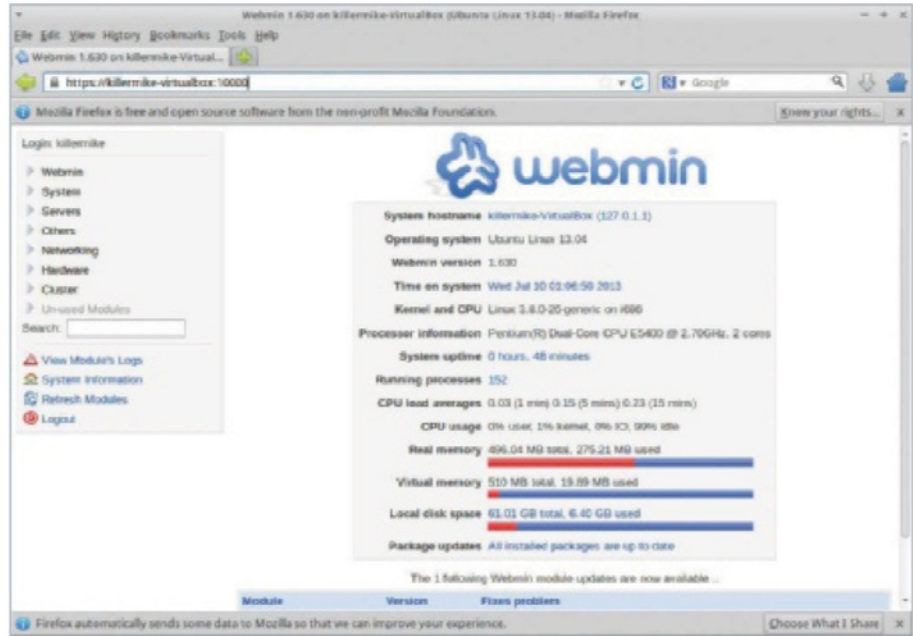
02 Install Webmin (for DEB)

In the case of Debian-derived distributions, like Ubuntu, install the DEB file by typing `sudo dpkg -i [name of .deb you downloaded]`. This won't satisfy all of the required dependencies for the package, so don't worry if it throws up some errors. To fix this, type: `sudo apt-get -f install`



03 Install Webmin (for RPM)

In the case of Red Hat-derived distros such as CentOS, first make yourself root by typing `su`. To install, type: `yum install [name of RPM file you've downloaded]`.



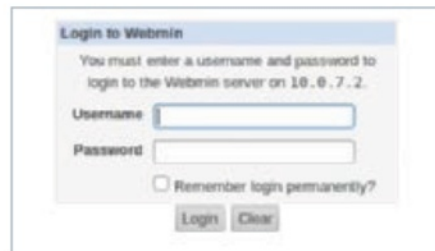
04 Connect to Webmin

Fingers crossed, Webmin is now working. You can test it by navigating to `https://[your hostname]:10000/`. You can discover your hostname by typing `hostname` at the command prompt. If everything is working, you should see the web interface for Webmin.

similar for a wired network; 'wlan0' for a Wi-Fi one). From here, change the IP address to something congruent to your current numbering scheme but higher. For example from 192.168.1.5 to 192.168.1.200.

05 Log in with root

Typically, you will give Webmin your 'root' username and password, when prompted. On a system such as Ubuntu that has no root account, you can use the username and password of the regular user so long as that user can execute sudo.

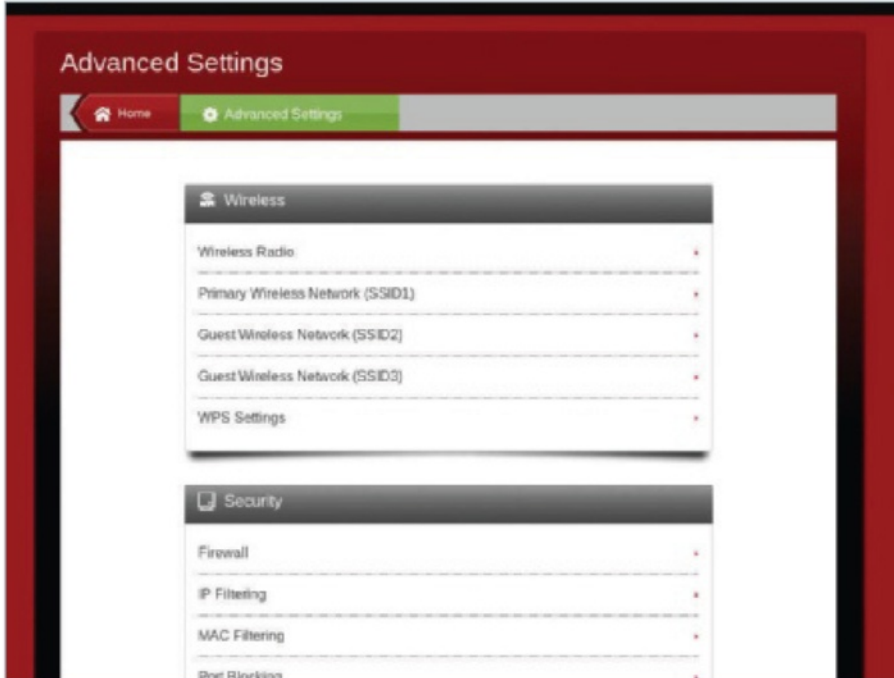


06 Set a static IP address

For convenience, you might want to assign a static IP address to the host machine, so that it can be consistently identified on the network. In Webmin, go to Networking>Network Configuration>Network Interfaces to see a list of currently activated network interfaces. Click on the name of the network adaptor used to connect to the rest of your network ('eth0' or

07 Configure your local firewall

If you can't access Webmin from other machines on your network (by using `http://[IP address:10000]/`), don't panic: it probably means a local firewall is blocking port 10000. You can configure the local firewall using Webmin itself, too. Go to Networking>Local Firewall and click on the add button. Most of the fields on the next page can be ignored, but set Network Protocol to equals TCP and Source TCP port to equals 10000. Click on Create to apply.

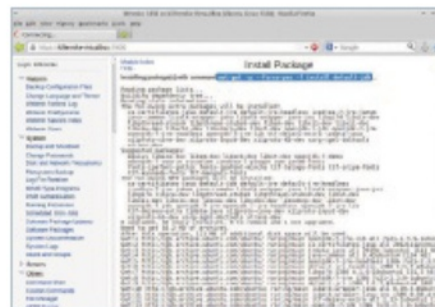


08 Configure your router How you enable access from the internet is specific for each router, so you'll need to examine the instructions for your router or visit portforward.com. In short, you need to allow incoming transmissions for port 10000 and forward that port to the IP address of the host computer on your network. This is usually very straightforward, but since every router is different, it's impossible to explain in detail here.

09 Webmin package management In this example, we'll install OpenJDK to a stock Ubuntu machine using the Webmin package management facilities. Open the System submenu from the sidebar and select Software Packages. Note that the Search... button opens a window that allows searching of the APT database.

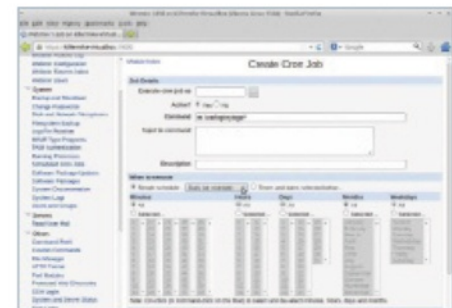


10 Package Installation We'll install Java support on Ubuntu in this example, but things work in much the same way on other platforms. On the main page, select the package from APT and enter default-jdk into the text box. Now select Install. Like many of the Webmin modules, this window contains quite a lot of useful text output. At the top it shows the Apt command that it shall execute (apt-get -y --force-yes -f install default-jdk in this case). Beneath this, it shows the output of Apt and below this, a tabular summary of all packages installed.

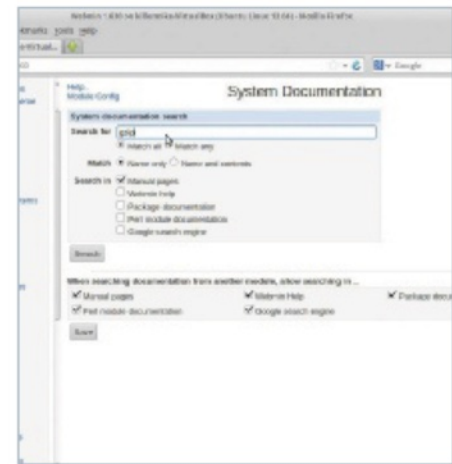


11 Manage cron jobs The syntax of specifying a new scheduled UNIX job, or cron job, is notoriously fiddly, and Webmin offers a neat front-end for this. Begin by entering the System>Scheduled Cron Jobs page. On this page, you can create new jobs or edit jobs already on the system.

12 Create a cron job We're going to create a nightly job that deletes the contents of /var/log/mylogs using the Scheduled Cron Jobs page. Select the 'Create a new scheduled cron job' option and this takes you to the cron job editing page. In the job details section, specify that the job will be executed by user root. Type `rm /var/log/mylogs/*` into the Command text box. Add a description for the job to the Description box. In the 'When to execute' section, select Simple schedule and Daily (at midnight). Click on Create.

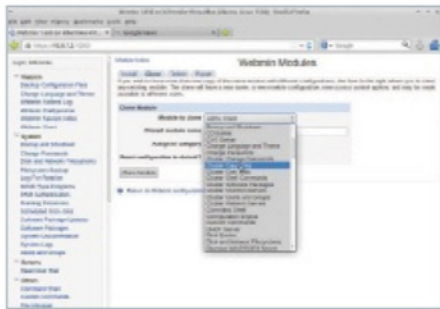


13 Search system documentation The System Documentation module (accessed from within the System section), offers a complete man page browser, usefully, accessing the man files on the host system. To use it, simply enter a search term. It operates much the same way as the standard 'man' command, but many will find the web-based interface more convenient.



14 View logs The Webmin Actions Log is a feather in Webmin's cap. If you don't add anything to the default form and click on search, you will be given a summary of configuration changes made that day. You can further refine your search by, for example, widening the date range.

15 Clone module
 Clone a module to have a second (or more) version with different settings. Go to Webmin> Webmin Configuration>Webmin Modules. Now select the Clone tab. From here, select a module to clone from the drop-down and give it a new name. Click on 'Clone module' and it will be added to the sidebar.



16 Execute commands
 Webmin has basic facilities for the execution of commands on the host machine, with a display of the output using the Command Shell page. If there's a command that you want to assign to a clickable button for frequent use, add it in the Custom Commands page.

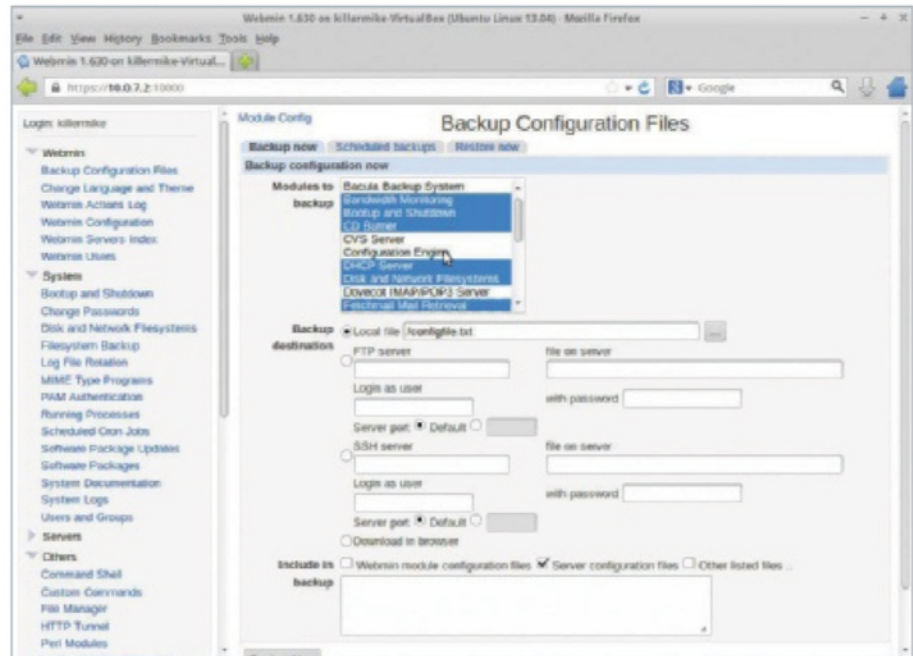


17 Text login
 If you need a bit more flexibility than the execute commands page allows, but you don't want to (or aren't able to) install SSH, try Others> Text Login. This offers a full terminal in which you can run text-mode programs, all from within the browser.

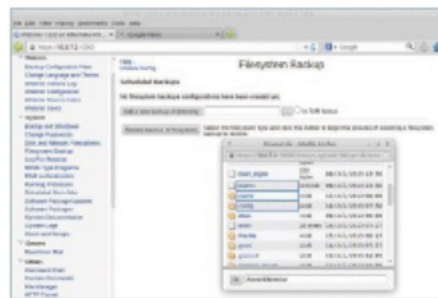


“The Webmin Actions Log is a serious feather in Webmin’s cap”

18 Back up config files
 You can back up any configuration that Webmin modules can access to a local or remote file for later restoration on the Webmin>Backup Configuration Files page. You can independently select which modules to back up and restore (Ctrl- or Shift-click), and there's a facility for scheduled backups.



19 File system backup
 Webmin includes a simple facility to back up files and directories. To use it, go to System>Filesystem Backup. Specify a directory and click on 'Add backup directory'. From here you can specify details such as a schedule for the backups and the remote (SSH, FTP) or local destination for the archive.



20 Monitor bandwidth
 Go to Networking>Bandwidth Monitoring and click on 'Set up now'. Once it's set up, you have to create some traffic on that interface before a report can be generated. When you've done this, specify a time range and click on 'Generate report'. Note that this function has a performance overhead.

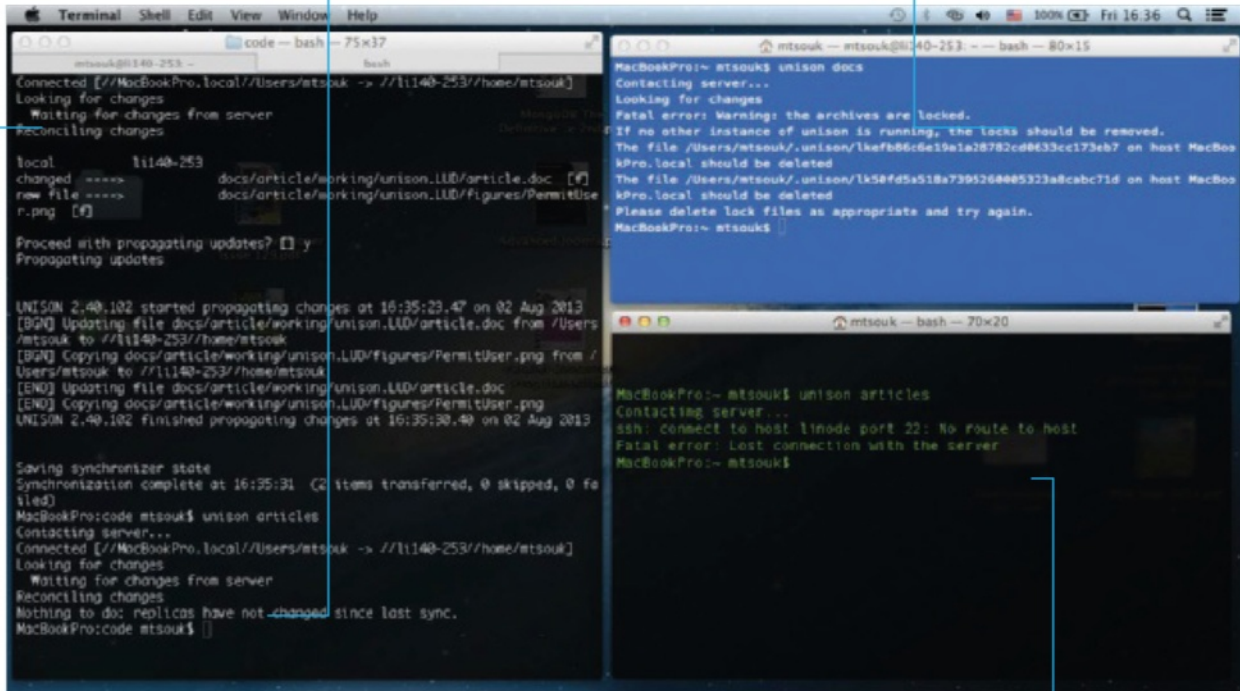
21 Add a disk-space monitor
 Webmin monitors alert you when certain conditions are met. Go to Others>System and Server Status. Click on Scheduled Monitoring and specify your SMTP email details. Select Disk Space from the drop-down and specify a 5% minimum size. Click on Create. Congratulations, you've started to master Webmin!



Unison synchronises files between computers – here we’re using the command-line version of the tool

This line indicates that Unison has nothing to sync at the moment, since no files have changed

This error message shows that there is a lock file that needs to be manually removed



The network connection is broken so Unison cannot synchronise files

Synchronise your files with Unison

Learn how to use the Unison command-line tool to synchronise files between computers quickly and reliably

Resources

Unison: www.cis.upenn.edu/~bcpierce/unison
Two networked UNIX machines

Unison is an open source file synchronisation tool for both text and binary files. It also has a GUI, but here we’re focusing on only the command-line version because it’s quicker and gets the job done cleanly. Unison really shows its capabilities when you are working with more than one computer and you need synchronisation across all of them.

Benjamin C Pierce led the creation of Unison at the University of Pennsylvania and it started life as a research project. It can be used through the

SSH service and works equally well on both UNIX (Linux, Mac OS X, etc) and Windows machines.

It should be apparent that Unison was inspired by the rsync utility. Unison differs from rsync in that the latter is a mirroring tool that needs to know in advance where the willing-to-keep versions of the files are, whereas Unison is a synchronisation tool that identifies the files that have been changed since the last sync process and decides the way that the changes are going to be propagated. In short, it’s smart.


```

1 # Mihalis Tsoukalos
2 #
3 # Friday 05 July 2013
4 #
5 root = /Users/mtsouk
6 root = ssh://linode//home/mtsouk
7
8 # Paths to synchronize
9 path = docs/article/working
10
11 # Log file
12 logfile = /Users/mtsouk/.unison/unison.log
13
14 # Backup files
15 backup = Name *
16 ignore = Name *.DS_Store
17 ignore = Name .DS_Store
18
19

```

05 Running Unison using a simple profile file

Some basic things you need to know about Unison profiles are:

- The default backup level is 2, which controls how many previous versions of each file are kept not including the current version of the file. So the default backup level keeps two backup copies of a file. The parameter for setting the backup level is called maxbackups.
- You can tell Unison not to keep file backups by not adding any backup options inside a Unison profile file.
- Profile files support Unicode characters.
- The logfile option tells Unison to keep a log file called unison.log inside the .unison directory.
- Sometimes, network problems prohibit file synchronisation, but Unison's error messages are denoting the problem:

```

$ unison articles
Contacting server...
Connection closed by 109.xxx.yyy.zzz
Fatal error: Lost connection with the server

```

06 Explaining the profile file

Lines starting with a # denote comments and are not processed any further. Exactly two lines must start with root =. They declare the machines (one is always the local

computer) that are going to participate in the syncing process as well as the directories that are considered the root directories for each machine. After those declarations, the actual directories that are going to be synced are listed. In this example we have just one directory (one path = line). Its full path is /Users/mtsouk/docs/article/working for the local machine and /home/mtsouk/docs/article/working for the machine called linode (an IP alias inside /etc/hosts). You can have as many path entries as you want. All files are synced except the ones that match the ignore option. You can also have as many ignore = lines as you want and can use regular expressions in them. The backup option tells Unison to keep backups of all files.

```

$ unison articles
Contacting server...
Connection closed by 109.xxx.yyy.zzz
Fatal error: Lost connection with the server

```

07 The .unison directory

Most of the Unison housekeeping happens inside the .unison directory which resides in the user's home directory.

Unison keeps the following data there:

- Its profile files including the default profile called default.prf.
- Its backup files, if you decided to have central backups.
- The status file for each used profile.
- The Unison log file, should you tell it to keep one. You can put it anywhere you want but it is very convenient for it to reside there.
- Unison creates lock files there, during synchronisations, which it deletes when finished. **Note:** It is not recommended to synchronise the whole .unison directory. Synchronising just the profiles is okay.

08 A more advanced Unison profile file

The contents of the advanced.prf file are as follows:

```

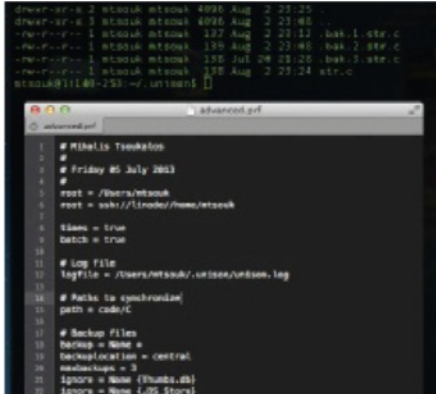
root = /Users/mtsouk
root = ssh://linode//home/mtsouk
times = true
batch = true
# Log file
logfile = /Users/mtsouk/.unison/unison.log
# Paths to synchronize
path = code/C
# Backup files
backup = Name *
backuplocation = central
maxbackups = 3
ignore = Name {Thumbs.db}
ignore = Name {.DS_Store}
ignore = Path {Samples.lnk}

```

```

$ unison articles
Contacting server...
Connection closed by 109.xxx.yyy.zzz
Fatal error: Lost connection with the server

```



09 Explaining the advanced profile file

- The `times = true` line tells Unison to synchronise modification times.
- The `maxbackups = 3` line tells Unison to keep the current file version plus three backups of it.
- The `backup = Name *` line tells Unison to back up every file.
- The `backuplocation = central`, which is the default option, tells Unison to keep all backups in a central location. If neither the `backupdir` preference nor the environment variable `UNISONBACKUPDIR` are set, the `.unison/backup` directory is used as the backup location. If set to `local`, then all backups will be kept in the same directory as the original files.
- The `batch = true` option is a little tricky and you should be careful with it as Unison will ask no questions at all and non-conflicting changes will be propagated whereas conflicts will be skipped. Nevertheless, it is an essential option if you want to use Unison as a cron job.
- The `ignore = Name {*.DS_Store}` line tells Unison to not synchronise files that end with `.DS_Store`.

10 Using SSH without giving a password

The single most time-saving thing you can do is to set up SSH so that you will not need to enter your password each time you want to synchronise your files and directories. The procedure is easy and involves the following three steps:

1. Run `ssh-keygen -t rsa`

You will have to enter a passphrase twice, so please do remember the passphrase! Two new files are going to be created: `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.

2. You may need to create a directory called `.ssh` on the remote server if it does not already exist.

3. Copy the contents of the `~/.ssh/id_rsa.pub` file from your local server into the file `~/.ssh/authorized_keys` found on the remote server. One way of doing it is by executing the following command:

```
$ cat ~/.ssh/id_rsa.pub | ssh linode 'cat >> .ssh/authorized_keys'
```

The next time you try to log into the remote Linux server using SSH, you will be asked for the passphrase of step 1 for the last time.

From now on, you can log into the remote Linux server by just typing `ssh linode`:

```
$ ssh linode
Linux (none) 3.9.3-x86_64-linode33
#1 SMP Mon May 20 10:22:57 EDT 2013
x86_64
.
.
.
Last login: Wed Jul 31 18:46:23 2013
from ppp-94-64-21-97.home.otenet.gr
mtsouk@11i140-253:~$
```

The first time you log into the remote server without typing your password, the following informative message will be on the screen:

```
Identity added: /Users/mtsouk/.ssh/
id_rsa (/Users/mtsouk/.ssh/id_rsa)
```



11 Two common Unison troubleshooting techniques

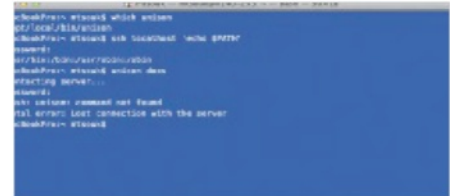
There are times when things do not work as expected. Unison offers you many options that can help you both find and solve problems.

The first option to try is the `-testserver` option that just connects to the remote server and then exits without synchronising any files.

The second thing to do is run the following command:

```
$ ssh remote.machine.domain 'echo $PATH'
```

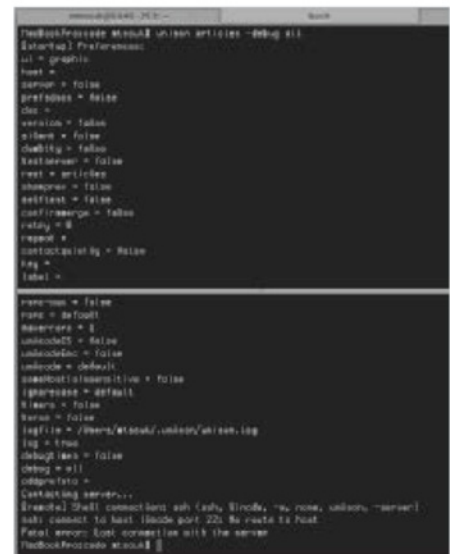
The aforementioned command let you see whether the `PATH` is the same as when you log in using `ssh remote.machine.domain`. If the problem is with the `PATH`, check if the option `PermitUserEnvironment` in `/etc/ssh/sshd_config` is set to `'no'` and change it to `'yes'`.



12 Unison hints and tips

The first two or three times you use a new profile, double-check if everything works as expected.

- You do not need to use every parameter that Unison supports, just the ones that will do your job!
- You can troubleshoot Unison using the `-debug` all command-line option. It will generate lots of output useful for debugging.
- The more you use Unison, the more you will understand its practicality.
- You should be very careful with your backup options, especially `maxbackups`, as it can take up too much space on your computer.
- You can use Unison to securely exchange files between computers.
- If a Windows machine is involved in the synchronisation process, be careful with file and directory permissions.
- For non-critical data files you may run Unison once a day, but for critical data you should run it more often.
- Unison cannot replace regular backups!
- When you are making a new profile, either start simple or use an existing one as a starting template. Add the extra functionality and features while making sure that you always have a working profile.

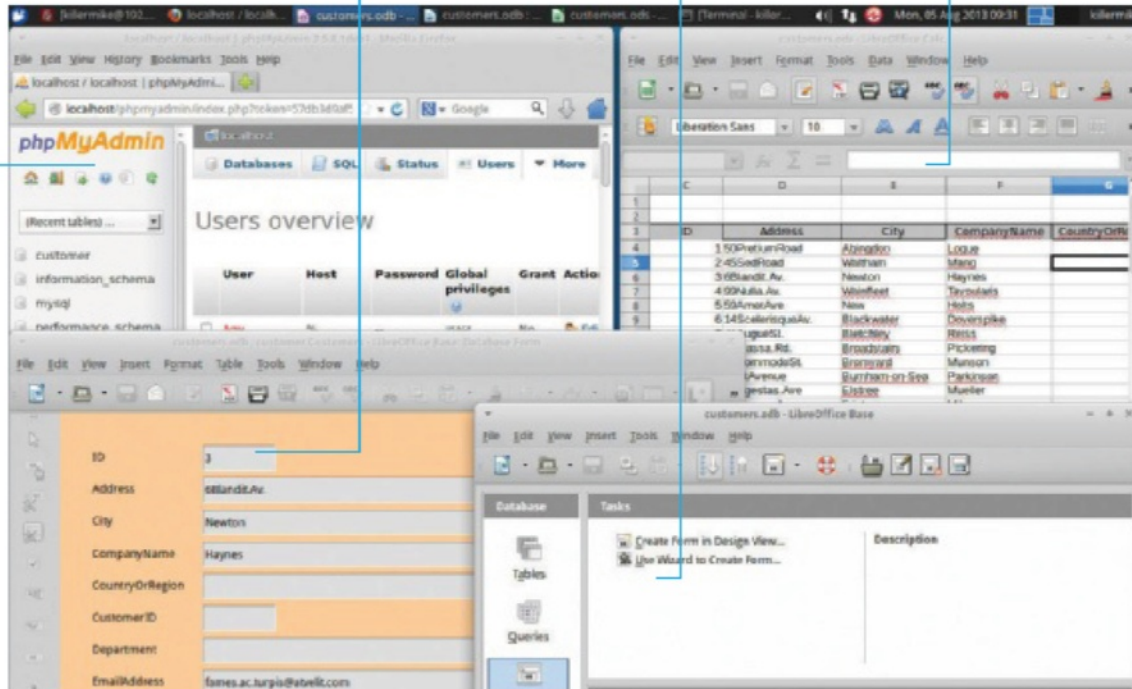


phpMyAdmin offers a web-based front-end for the creation and maintenance of MySQL databases

Data entry is carried out via an easy-to-use form in Base, the front end to our database

All of the actual database design (fields, form layout etc) is carried out from within Base

You import and export data to and from LibreOffice Base by using Calc, the spreadsheet module. This enables access to most common data formats



Make a small business database with LibreOffice

Create a database that combines an easy-to-use, form-based front-end using LibreOffice with a portable, networked MySQL back-end

We're going to show you how to put together a typical database for small business use: a database of customer details. It will be possible to both export and import contact data in standard formats by making use of Calc, LibreOffice's spreadsheet module. We'll use Gmail contacts as our source, but you can use any software that can export CSV files – and pretty much everything can.

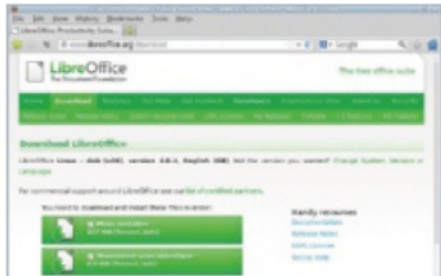
We've added a few twists to keep things interesting. This project uses the Base module of LibreOffice as the front-end, and this provides a GUI for setting up the database, creating the

forms for data entry and the actual business of entering data. For the back-end, we will be using the industry-standard MySQL. This allows us to locate the back-end on a central server. This, in turn, allows multiple users to access the database.

For initial creation of the MySQL database, we'll use phpMyAdmin thanks to its friendly web interface, although the actual database design will be carried out from within Base. By the end of the project, you will have a GUI system for browsing and editing the database with a portable, networked back-end.

Resources

LibreOffice: www.libreoffice.org
At least one Ubuntu Linux PC



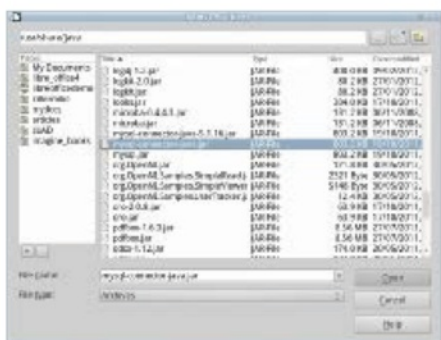
01 Install LibreOffice

At time of writing, the major Linux distributions haven't moved over to LibreOffice 4 and are still offering 3.x. This means that you may have to install LibreOffice 4 manually. Visit the LibreOffice website (www.libreoffice.org) and follow the instructions. On Ubuntu, this consists of unpacking the archive and running `sudo dpkg -i *.deb` on the contents.



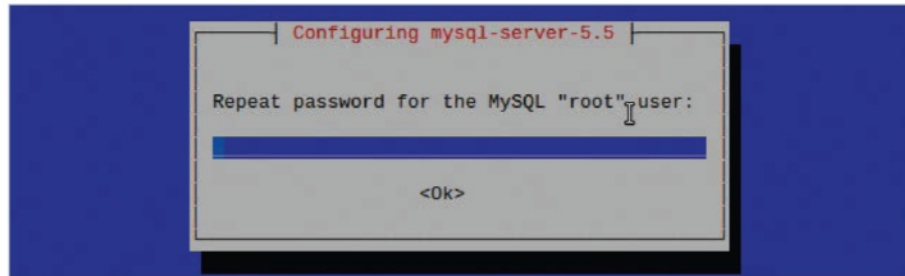
02 Install the Apache web server

We'll install Apache early on and with its own command because some of the other packages need to be able to configure a working installation. Carry out the installation with `sudo apt-get install apache2`. Test it by navigating to `http://localhost`.



03 Install Java and additional classes

Connectivity between Base and MySQL makes use of a Java class. Type `sudo apt-get install default-jdk` to install the Java runtime. Type `sudo apt-get install libmysql-java` to install the needed additional Java classes.



04 Install the MySQL Server

Type `sudo apt-get install mysql-server` to begin installation. Before long, you should be prompted to set a root MySQL password. Note this isn't the same as the administrator account of your system, which is also called 'root'. Choose a password and make a note of it.

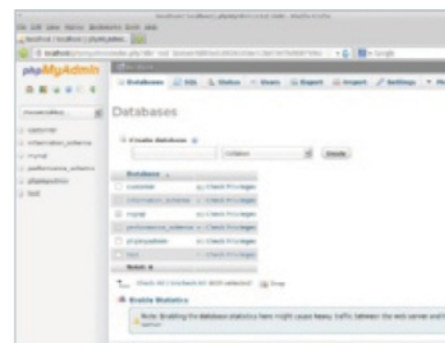
05 Install and test phpMyAdmin

Type `sudo apt-get install phpmyadmin` to begin installation. When prompted to choose a web server, choose Apache2, select it with the space bar and press Return. When requested, give it the MySQL root password and then choose a password for phpMyAdmin and make a note of it. Navigating to `http://localhost/phpmyadmin/` should take you to a functioning login page. Log in using the MySQL root username and password. We'll use MySQL to set up and maintain the actual database, although we'll create the fields from within LibreOffice later on.



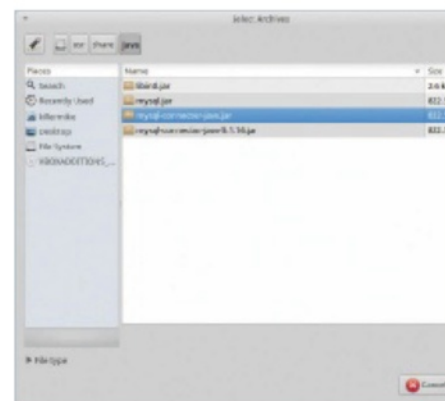
06 Create database

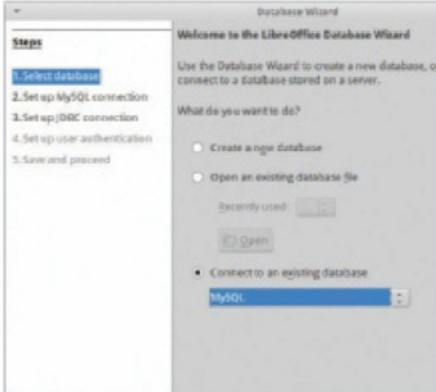
Within the phpMyAdmin web interface, select the Databases tab. Now create a new database by entering the name 'customer' into the text box and clicking on Create. This database will contain our customer data.



07 Add JDBC in LibreOffice

We now need to tell LibreOffice where to find the JDBC class file. Start LibreOffice and go to LibreOffice>Options>Advanced. In the Java Options section, select Class Path and then Add Archive. The file you need is located at: `/usr/share/java/mysql-connector-java.jar`. Select it and restart LibreOffice.



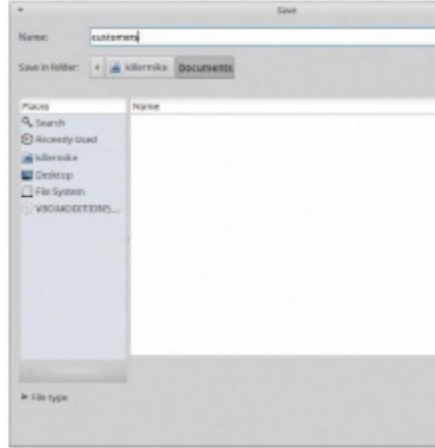


08 Connect the database We now need to connect our front-end (LibreOffice) to the back-end (MySQL) of our database. Start LibreOffice and launch the Base module. In the dialog that pops up, select 'Connect to an existing database'. From the drop-down menu below this, select MySQL as the database type.

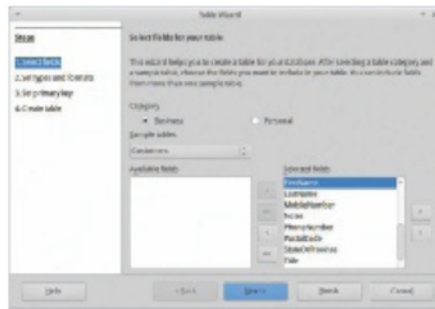


09 Configure Base On the next page, select 'Connect using JDBC'. On the next page, click on 'Test class' to ensure that the Java RT is working. Now enter the name of the database that we created, customer, and enter localhost into the Server field. On the next page, add Root as the username and tick 'Password required'. Now click on the 'Test connection' button and enter the root MySQL password, when asked for it, to test the connection between LibreOffice and the local MySQL server. Presuming that this completes without errors, click on Next.

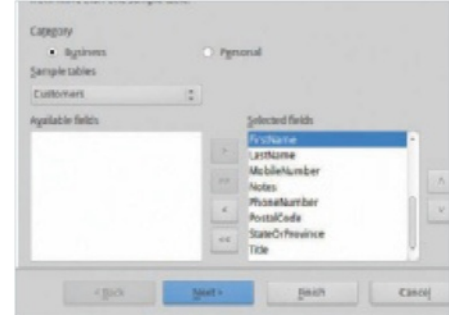
“Remember that this file contains the connection information for access to our MySQL database – it doesn’t contain the actual records”



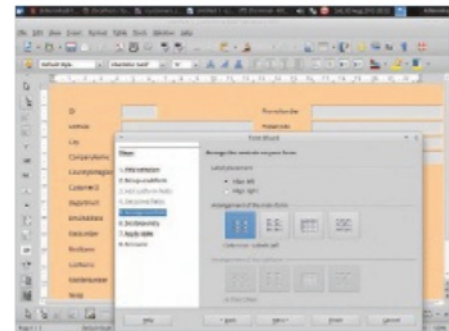
10 Save the database You can accept the defaults on the next page, so click on Finish. When prompted, give the database a name and save it. Remember that this file contains the connection information for access to our MySQL database – it doesn’t contain the actual records.



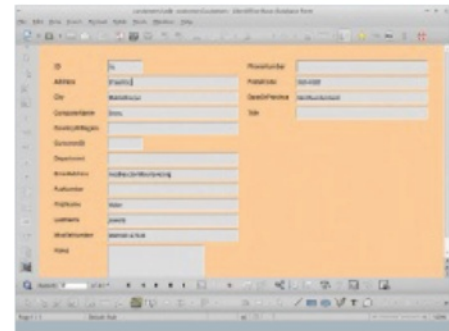
11 Create fields from a template You may want to create a custom set of fields for your customer records, but to save time we’re going to use the one of the templates that is built into Base. Select Tables from the side menu and then 'Use Wizard to Create Table'. Using the Sample tables pull-down menu, select Customers. Use the >> button to copy all of them across. On the next page, you can tweak the fields that you have included and add new ones. Select the defaults on the next two pages and then click on Finish.



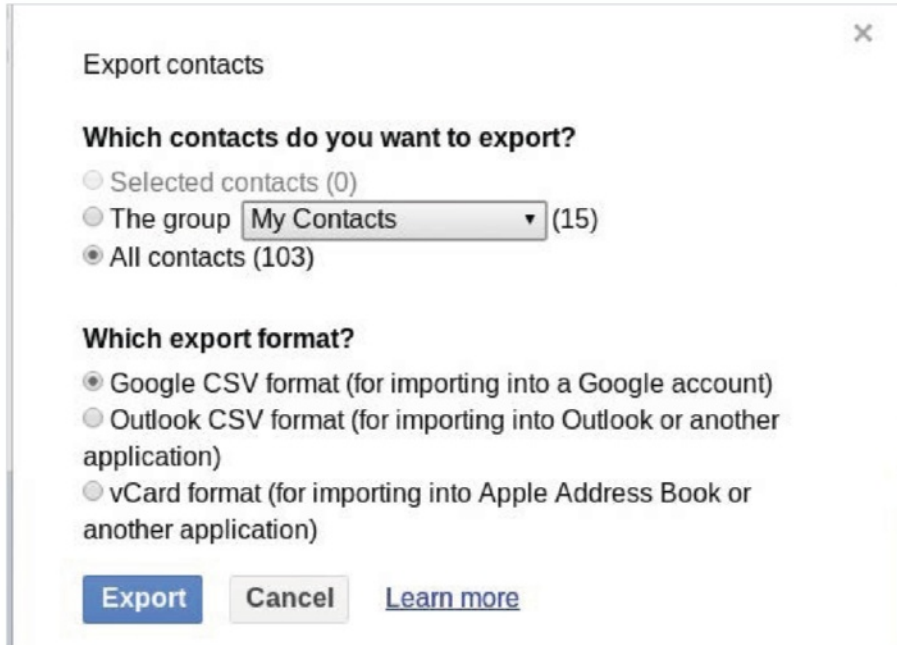
12 Create form from template Select Forms from the sidebar. Click on 'Use Wizard to Create Form' in the Tasks window. In the table wizard, click on the >> symbol to copy across all of the fields in the database.



13 Finalise form Accept the defaults in sections 2, 3 and 4, but select the first arrangement icon in section 5, 'Arrange controls'. You should now see a preview of our entry form in the main window. Select defaults on the other sections and then click on Finish.

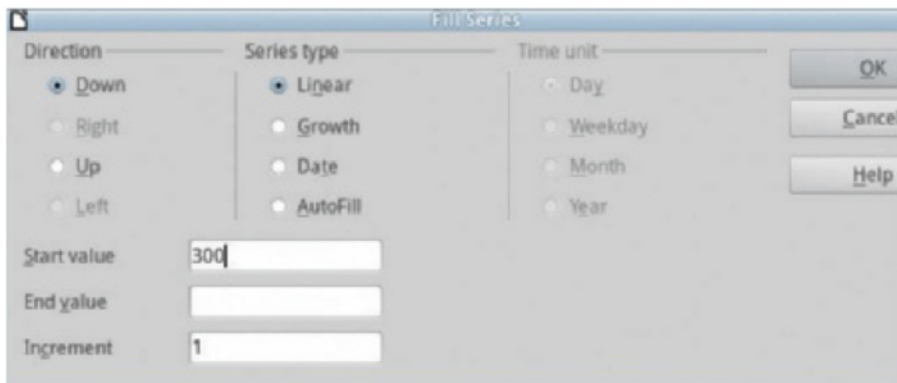


14 Test data entry To enter data into the database, use the form that we created. Select Forms in the sidebar and then double-click on the name of the form in the main window. This brings up the GUI record-editing interface. The form can still be tweaked and edited by right-clicking on its name in the main window.



15 Export your contacts from Gmail
Switch from the Gmail contacts view using the pull-down menu in the top-left corner, underneath the Gmail logo. From here, click on the More icon pull-down menu and select Export... Click on Export.

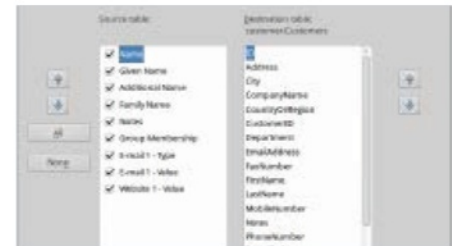
16 Clean up the data and create a key
Start a new spreadsheet and open the CSV file that you exported from Gmail. Use Ctrl-mousewheel zooming to get an overview. Typically, a lot of the fields will be completely blank, so select these columns (click on the column letter at the top of the window) and remove them (Edit>Delete cells). We have to create a key for each record. Label a column ID. Select the first cell in the column and then select the final cell by Shift-clicking on it. Use the fill feature (Edit>Fill>Series).



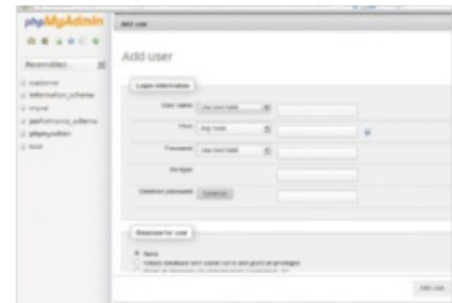
17 Import the data into Base
When you've cleaned up the spreadsheet, select the data (including the column headers) by clicking on the top-left cell and then Shift-clicking on the bottom-right cell. Right-click and select Copy. Select Tables from the side menu of the Base module. From here, right-click on the customers table and click on Paste. This should bring up the import wizard. Select 'Append data' and 'Use first line as column names' options, and click on Next.



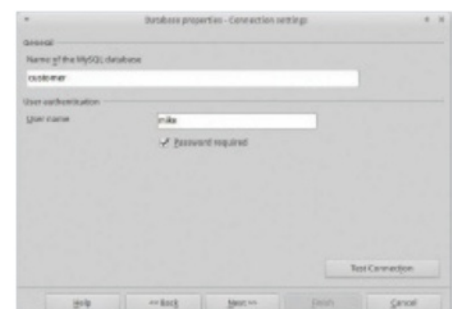
18 Align the fields
The field names from our imported data don't quite match those of the database and so we need to use the second page of the wizard to line them up. To do so, click on a field name and use the up and down icons in the other list to create the correct attachments. Then click on Create.



19 Create a new database user
To access the database from more than one machine, you must create additional users. Log back into phpMyAdmin, click on the Users tab and select 'Add user'. From here, create a new user with the name and password of your choosing and make a note of it. Click on 'Check all' in the 'Global privileges' section.



20 Redistribute the database
In the Base module, re-save the database under a new name. In this new version of the file, we have to alter a few details. Select Edit>Database>Properties and enter the name of the new database user. Click on the Additional Settings tab and enter the IP address of the machine with the MySQL database.



The replica set consists of nodes 192.168.2.4 (port 27019), 192.168.1.10 (port 27019) and 192.168.2.3 (port 27018)

```

Mon Jul 1 11:09:56.388 [rsStart] trying to contact 192.168.2.3:27018
Mon Jul 1 11:09:56.397 [rsStart] replSet I am 192.168.2.4:27019
Mon Jul 1 11:09:56.397 [rsStart] replSet got config version 1 from a remote, saving locally
Mon Jul 1 11:09:56.397 [rsStart] replSet info saving a newer config version to local.system.replset
Mon Jul 1 11:09:56.415 [rsStart] replSet saveConfigLocally done
Mon Jul 1 11:09:56.415 [rsStart] replSet STARTUP2
Mon Jul 1 11:09:56.428 [rsSync] *****
Mon Jul 1 11:09:56.433 [rsSync] creating replication oplog of size: 192MB...
Mon Jul 1 11:09:56.434 [FileAllocator] allocating new datafile ./mongo5/local.1, filling with zeroes...
Mon Jul 1 11:09:56.434 [FileAllocator] creating directory ./mongo5/_tmp
Mon Jul 1 11:09:57.730 [FileAllocator] done allocating datafile ./mongo5/local.1, size: 256MB, took 1.296 secs
Mon Jul 1 11:09:57.842 [rsSync] *****
Mon Jul 1 11:09:57.842 [rsSync] replSet initial sync pending
Mon Jul 1 11:09:57.842 [rsSync] replSet initial sync need a member to be primary or secondary to do our initial sync
Mon Jul 1 11:09:58.399 [rsHealthPoll] replSet member 192.168.1.10:27019 is up
Mon Jul 1 11:09:58.400 [rsHealthPoll] replSet member 192.168.2.3:27018 is up
Mon Jul 1 11:09:58.400 [rsHealthPoll] replSet member 192.168.2.3:27018 is now in state SECONDARY
Mon Jul 1 11:10:00.102 [initandlisten] connection accepted from 192.168.2.4:61515 #2 (2 connections now open)
Mon Jul 1 11:10:00.102 [conn2] end connection 192.168.2.4:61515 (1 connection now open)
Mon Jul 1 11:10:00.103 [initandlisten] connection accepted from 192.168.2.4:61516 #3 (2 connections now open)
Mon Jul 1 11:10:00.400 [rsHealthPoll] replSet info 192.168.1.10:27019 thinks that we are down
Mon Jul 1 11:10:00.400 [rsHealthPoll] replSet member 192.168.1.10:27019 is now in state STARTUP2
Mon Jul 1 11:10:06.159 [conn1] replSet RECOVERING
Mon Jul 1 11:10:06.159 [conn1] replSet info voting yea for 192.168.2.3:27018 (3)
Mon Jul 1 11:10:06.403 [rsHealthPoll] replSet member 192.168.1.10:27019 is now in state RECOVERING
Mon Jul 1 11:10:08.415 [rsHealthPoll] replSet member 192.168.2.3:27018 is now in state PRIMARY
Mon Jul 1 11:10:13.843 [rsSync] replSet initial sync pending
Mon Jul 1 11:10:13.843 [rsSync] replSet syncing to: 192.168.2.3:27018
Mon Jul 1 11:10:14.024 [rsSync] build index local.me { _id: 1 }
    
```

Which is the primary node

The replica set is recovering

Synchronising data to node 192.168.2.3

Create and save data with a MongoDB database

Forget about joins and SQL and try NoSQL databases – specifically MongoDB, the leading example

Resources

MongoDB: www.mongodb.org

Pymongo: api.mongodb.org/python/current/

MongoDB is an open source document-oriented database system written in C++ by Dwight Merriman and Eliot Horowitz. It runs on UNIX machines as well as Windows and supports replication and sharding (aka horizontal partitioning) – the process of separating a single database across a cluster of machines.

Many programming languages – including C, C++, Erlang, Haskell, Perl, PHP, Python, Ruby

and Scala – support MongoDB. It is suitable for many things, including archiving, event logging, storing documents, agile development, real-time statistics and analysis, gaming, and mobile and location services.

This article will show you how to store Apache log files in a MongoDB database with the help of a small Python script, which can be found at <http://bit.ly/Kmva1v>.


```
monastery:~ mtsouk$ mongo --version
MongoDB shell version: 2.4.4
monastery:~ mtsouk$ mongo
MongoDB shell version: 2.4.4
connecting to: test
>
```

01 Connecting to MongoDB for the first time

Your Linux distribution probably includes a MongoDB package, so go ahead and install it. Alternatively, you can download a precompiled binary or get the source code from www.mongodb.org and compile it yourself.

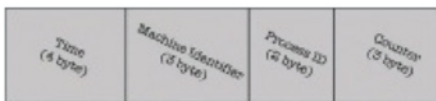
After installation, type `mongo --version` to find out the MongoDB version you are using and `mongo` to run the MongoDB shell and check if the MongoDB server process is running.

SQL Term	MongoDB Term
Database	Database
Table	Collection
Index	Index
Row	BSON document
Column	BSON field
Primary Key	_id field
Group by	Aggregation
Join	Embedding and Linking

02 MongoDB terminology

NoSQL databases are designed for the web and do not support joins, complex transactions and other features of the SQL language. You can update a MongoDB database schema without downtime, but you should design your MongoDB database without joins in mind.

Their terminology is a little different from the terminology of relational databases and you should familiarise yourself with it.



03 The _id field

Every time you insert a BSON document in MongoDB, MongoDB automatically generates a new field called `_id`. The `_id` field acts as the primary key and is always 12 bytes long. To find the creation time of the object with `_id` `'51cb590584919759671e4687'`, execute the following command from the MongoDB shell:

```
> ObjectId("51cb590584919759671e4687").getTimestamp()
ISODate("2013-06-26T21:11:33Z")
```

Note: You should remember that queries are case-sensitive.

```
64.242.88.10 - - [07/Mar/2004:16:05:49 -0800] "GET /twiki/bin/edit/Main/Double?topicparent=Main.ConfigurationVariables HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:16:06:51 -0800] "GET /twiki/bin/rdiff/TWiki/NewM rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision 200 6291
64.242.88.10 - - [07/Mar/2004:16:11:58 -0800] "GET /twiki/bin/view/TWiki/WikiS .1" 200 7352
64.242.88.10 - - [07/Mar/2004:16:20:55 -0800] "GET /twiki/bin/view/Main/DCCAnd /1.1" 200 5253
64.242.88.10 - - [07/Mar/2004:16:23:12 -0800] "GET /twiki/bin/oops/TWiki/Appen m?template=oopsmore&param1=1.12&param2=1.12 HTTP/1.1" 200 11382
64.242.88.10 - - [07/Mar/2004:16:24:16 -0800] "GET /twiki/bin/view/Main/PeterT .1" 200 4924
64.242.88.10 - - [07/Mar/2004:16:29:16 -0800] "GET /twiki/bin/edit/Main/Header. cparent=Main.ConfigurationVariables HTTP/1.1" 401 12851
64.242.88.10 - - [07/Mar/2004:16:30:29 -0800] "GET /twiki/bin/attach/Main/Offic HTTP/1.1" 401 12851
64.242.88.10 - - [07/Mar/2004:16:31:48 -0800] "GET /twiki/bin/view/TWiki/WebTop ate HTTP/1.1" 200 3732
```

04 Inserting an Apache log file into MongoDB

Now that you know some things about MongoDB, it is time to do something interesting and useful. A log file from Apache will be inserted inside a MongoDB database using a Python script.

```
The Python script is executed as follows:
$ zcat www6.ex000704.log.gz | python2.7 storeDB.py
```

...where `www6.ex000704.log.gz` is the name of the compressed (for saving disk space) log file.

```
1 #!/usr/bin/python
2 # Date: Wednesday 26 June 2013
3 #
4 # Description: This Python script reads an Apache log file,
5 # parses it and stores it in a MongoDB database
6 #
7
8 import sys
9 import pymongo
10 import re
11
12 # The number of BSON documents written
13 total = 0
14
15 # Open the MongoDB connection
16 connMongo = pymongo.Connection("mongodb://localhost:27017")
17 # Connect to database named LUD (Linux User Download)
18 db = connMongo.LUD
19 # Select the collection to save the log file data
20 logs = db.apacheLogs
21
22 # Read the file from stdin, line by line
23 for line in sys.stdin:
24     line = line.rstrip("\n")
25     parsed = re.findall(r'^(\d+)?\[(\d+)?\](\d+)?$', line)
26     # print parsed
27     total = total + 1
28     # Construct the log entry to be inserted
29     log = {
30         'host': parsed[0],
31         'date': parsed[1],
32         'document': parsed[2],
33         'statusCode': parsed[3],
34         'size': parsed[4]
35     }
36     # Store it!
37     log._id = logs.insert(log)
38     print "The %s of the inserted post is", log._id
39
40 # Close the MongoDB connection
41 connMongo.close()
42
43 # Present the total number of BSON documents written
44 print "Total number of documents written:", total
```

“MongoDB is supported by many programming languages”

05 The storeDB.py Python script

The `storeDB.py` script uses the PyMongo Python module to connect to MongoDB. The MongoDB server is running on localhost and listens to port 27017. For every inserted BSON document, its `_id` field is printed on screen. Finally, the script prints the total number of documents inserted in the MongoDB database.

The host and its port number are hard-coded inside the script, so change them to match yours.

06 Connecting to MongoDB using PyMongo

You first need to connect to MongoDB using:

```
connMongo = pymongo.Connection('mongodb://localhost:27017')
```

You then select the database name you want (LUD) using the following line of code:

```
db = connMongo.LUD
```

And finally you select the name of the collection (`apacheLogs`) to store the data:

```
logs = db.apacheLogs
```

After finishing your interaction with MongoDB you should close the connection as follows:

```
connMongo.close()
```

```

> db.apacheLogs.find()
{ "_id" : ObjectId("51cb590584919759671e4674"), "statusCode" : "401", "document" : "wiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:05:49 -0800", "size" : "12846" }
{ "_id" : ObjectId("51cb590584919759671e4675"), "statusCode" : "200", "document" : "wiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:06:51 -0800", "size" : "4523" }
{ "_id" : ObjectId("51cb590584919759671e4676"), "statusCode" : "200", "document" : "dilman/listinfo/hsdivision HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:07:02 -0800", "size" : "6291" }
{ "_id" : ObjectId("51cb590584919759671e4677"), "statusCode" : "200", "document" : "wiki/bin/view/TWiki/WikiSyntax HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:11:58 -0800", "size" : "7352" }
{ "_id" : ObjectId("51cb590584919759671e4678"), "statusCode" : "200", "document" : "wiki/bin/view/Main/DCCAndPostFix HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:20:55 -0800", "size" : "5253" }
{ "_id" : ObjectId("51cb590584919759671e4679"), "statusCode" : "200", "document" : "wiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&param1=1.12&param2=1.12 HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:23:12 -0800", "size" : "11382" }
{ "_id" : ObjectId("51cb590584919759671e467a"), "statusCode" : "200", "document" : "wiki/bin/view/Main/PeterThoeny HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:24:16 -0800", "size" : "4924" }
{ "_id" : ObjectId("51cb590584919759671e467b"), "statusCode" : "401", "document" : "wiki/bin/edit/Main/Header_checks?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:29:16 -0800", "size" : "12851" }
{ "_id" : ObjectId("51cb590584919759671e467c"), "statusCode" : "401", "document" : "wiki/bin/attach/Main/OfficeLocations HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:30:29 -0800", "size" : "12851" }
{ "_id" : ObjectId("51cb590584919759671e467d"), "statusCode" : "200", "document" : "wiki/bin/view/TWiki/WebTopicEditTemplate HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:31:48 -0800", "size" : "3732" }
{ "_id" : ObjectId("51cb590584919759671e467e"), "statusCode" : "200", "document" : "wiki/bin/view/Main/WebChanges HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:32:50 -0800", "size" : "40520" }
{ "_id" : ObjectId("51cb590584919759671e467f"), "statusCode" : "401", "document" : "wiki/bin/edit/Main/Smtpd_etm_restrictions?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:33:53 -0800", "size" : "12851" }
{ "_id" : ObjectId("51cb590584919759671e4680"), "statusCode" : "200", "document" : "dilman/listinfo/business HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:34:19 -0800", "size" : "6379" }
{ "_id" : ObjectId("51cb590584919759671e4681"), "statusCode" : "200", "document" : "wiki/bin/view/Main/About HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:35:11 -0800", "size" : "40520" }

```

07 Displaying BSON documents from the apacheLogs collection

Type the following in order to connect to the MongoDB shell:

```
$ mongo
```

Select the desired database as follows:

```
> use LUD
```

See the available collections for the LUD database as follows:

```
> show collections
```

```
apacheLogs
```

```
system.indexes
```

Lastly, execute the following command to see all the contents of the apacheLogs collection:

```
> db.apacheLogs.find()
```

If the output is long, type 'it' to go to the next screen.

08 A replication example

Imagine that you have your precious data on your MongoDB server and there is a

power outage. Can you access your data? Is your data safe?

To avoid such difficult questions, you can use replication to keep your data both safe and available. Replication also allows you to do maintenance tasks without downtime and have MongoDB servers in different geographical areas.

09 Running the three MongoDB servers from the command line

For this example, you need three MongoDB server processes running.

We ran the three MongoDB servers, on their respective machines, as follows:

```

$ mongod --port 27018 --bind_ip 192.168.1.10 --dbpath ./mongo10 --rest --replSet LUDev
$ mongod --port 27019 --bind_ip 192.168.2.6 --dbpath ./mongo6 --rest --replSet LUDev

```

```

$ mongod --port 27018 --bind_ip 192.168.2.5 --dbpath ./mongo5 --rest --replSet LUDev

```

Note: You are going to see lots of output on your screen.

10 More information about the three MongoDB servers

You should specify the name of the replica set (LUDev) when you start the MongoDB server and have the data directory, specified by the --dbpath parameter, already created. You do not necessarily need three discrete Linux machines. You can use the same machine (IP address) as long as you are using different port numbers and directories.

```

> rs.initiate(
  { _id: 'LUDev', members: [
    { _id: 1, host: '192.168.1.10:27018' },
    { _id: 2, host: '192.168.2.6:27019' },
    { _id: 3, host: '192.168.2.5:27018' } ] })
{
  'info': 'Config now saved locally. Should be
  'ok': 1
}
>

```

11 The rs.initiate() command

Once you have your MongoDB server processes up and running, you should run the rs.initiate() command to actually create and enable the replica set.

If everything is okay, you will see similar output on your screen. If the MongoDB server processes are successfully running, most errors come from misspelled IPs or port numbers. The rs.initiate() command is simple but has a huge impact!

12 Information about replication

- Any node can be **primary**, but only one node can be primary at a given time.
- All write operations are executed at the primary node.
- Read operations go to primary and optionally to a secondary node.
- MongoDB performs automatic failover.
- MongoDB performs automatic recovery.
- Replication is not a substitute for backup, so you should not forget to take backups.

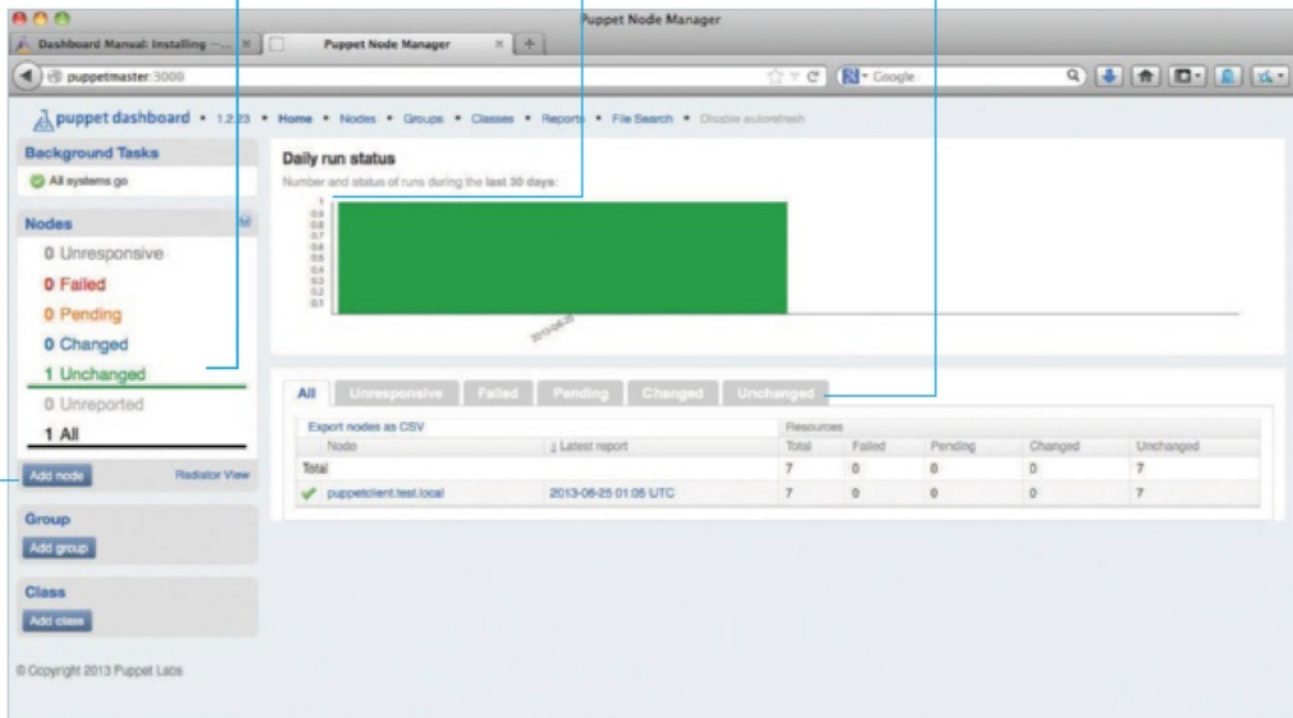
“Replication is not a substitute for backup”

Manually add nodes and groups using the buttons on the left-hand side

Puppet dashboard showing daily status with our node showing compliant

Stats on the left give a quick overall health status

Using the tabs, you can filter all the machines and review all those in a particular state



Maintain and manage all of your machines with Puppet

Set up your machines to be configured using Puppet so you can keep them in a consistent and workable state

Resources

Two networked servers

Ubuntu 12.04: www.ubuntu.com

Static IPs and full DNS entries

Keeping track of two machines, keeping them in sync is quite easy – for example, repo files and config files. However, once you start scaling past a machine or two, keeping files aligned over tens, hundreds and thousands of machines, it becomes a nightmare. If there is one thing a network manager likes, it is configuration standards.

This is where Puppet comes in. Puppet allows users to use extend control over the

contents of their files and keep them in sync across your estate. In this how-to guide to implementing a basic Puppet setup, we show how to keep all your files in sync.

This tutorial covers the basics of creating a basic Puppet server and client setup, through to setting up a few sample configurations that can be deployed, applying different configurations to different machines and configuring to clients in a standard manner.

```

stu@puppetmaster: ~ -- ssh -- 80x24
urreadahead will be reprofiled on next reboot
Setting up libreadline5 (5.2-11) ...
Setting up Augeas-lenses (0.10.0-0ubuntu4) ...
Setting up debconf-utils (1.5.42ubuntu1) ...
Setting up libruby1.8 (1.8.7.352-2ubuntu1.2) ...
Setting up ruby1.8 (1.8.7.352-2ubuntu1.2) ...
update-alternatives: using /usr/bin/ruby1.8 to provide /usr/bin/ruby (ruby) in a
via mode.
Setting up facter (1.6.5-1ubuntu1) ...
Setting up libaugeas0 (0.10.0-0ubuntu4) ...
Setting up libaugeas-ruby1.8 (0.3.0-1.1ubuntu4) ...
Setting up libruby (4.0) ...
Setting up libshadow-ruby1.8 (1.4.1-0build1) ...
Setting up puppet-common (2.7.11-1ubuntu2.3) ...
Setting up puppetmaster-common (2.7.11-1ubuntu2.3) ...
+ Starting puppet queue
...done.
Setting up puppetmaster (2.7.11-1ubuntu2.3) ...
+ Starting puppet master
...done.
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
stu@puppetmaster:~$ sudo touch /etc/puppet/manifests/site.pp
stu@puppetmaster:~$
    
```

```

stu@puppetclient: ~ -- ssh -- 80x24
Setting up libruby (4.0) ...
Setting up libshadow-ruby1.8 (1.4.1-0build1) ...
Setting up Augeas-lenses (0.10.0-0ubuntu4) ...
Setting up libaugeas0 (0.10.0-0ubuntu4) ...
Setting up libaugeas-ruby1.8 (0.3.0-1.1ubuntu4) ...
Setting up facter (1.6.5-1ubuntu1) ...
Setting up puppet-common (2.7.11-1ubuntu2.3) ...
Setting up puppet (2.7.11-1ubuntu2.3) ...
+ Starting puppet agent

puppet not configured to start, please edit /etc/default/puppet to enable
...done.
Setting up debconf-utils (1.5.42ubuntu1) ...
Setting up ruby (4.0) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
stu@puppetclient:~$ sudo puppetd --server puppetmaster.test.local --waitforcert
60 --test
Info: Creating a new SSL key for puppetclient.test.local
Info: Caching certificate for ca
Info: Creating a new SSL certificate request for puppetclient.test.local
Info: Certificate Request fingerprint (md5): 96:1D:C6:CC:82:2E:2D:25:3D:79:85:PS
:3E:69:67:83
    
```

01 Set up the Puppet master
 Puppet comes in two parts – master and agent nodes. The master node, as the name implies, is in charge. This server holds all the config file goodness (also known as manifests). For this tutorial we are running Ubuntu 12.04 LTS. Installing Puppet is really straightforward. Choose one of the hosts and install the Puppet master. Type the command:

```
sudo apt-get install puppetmaster
```

The setup requires the file `site.pp` to be present (more on what it is later). Do this by using:

```
sudo touch /etc/puppet/manifests/site.pp
```

This installs all the prerequisites of the server.

02 Set up the Puppet agent
 The agents sit on the machines that we want to effectively manage. To install all the components, use the command:

```
sudo apt-get install puppet
```

Again, this installs all the requirements for the agent or client. It is suggested that you do not set Puppet to autostart on boot. If you do this, by default the agent will contact the Puppet master and update its configuration, if needed, every 30 minutes. We are going to run ours manually, so that there is no waiting to see the changes take effect.

03 Configure the Puppet infrastructure

The next step is to set up the secure communication between the servers. To do this, log into the Puppet agent server and issue the command:

```
sudo puppetd --server puppetmaster.test.local --waitforcert 60 --test
```

You will have to edit the server name to reflect your setup. Leave the `--test` switch on as it'll show what is happening in the foreground, making life easier if there's a need to debug.

If you run the command and you get an error 'warning: Could not retrieve fact fqdn', it means you have not set up your DNS properly. It is strongly recommended that this is fixed before proceeding.

```

stu@puppetmaster: ~ -- ssh -- 80x
stu@puppetmaster:~$ sudo puppetca --list
"puppetclient.test.local" (56:1D:C6:CC:82:2E:2D:25:3D:79:85:PS:3E:69:67:83)
stu@puppetmaster:~$ sudo puppetca --sign puppetclient.test.local
notice: Signed certificate request for puppetclient.test.local
notice: Removing file Puppet::SSL::CertificateRequest pu
~/var/lib/puppet/ssl/csr/requests/puppetclient.test.local
stu@puppetmaster:~$
    
```

04 It's good to talk SSL
 The next step is to enable secure communications between the master and agent.

Type `sudo puppetca --list`. This will show all the client machines that are trying to connect to the server to service their requests. In order for them to be given access, we must allow them to do so, using the command:

```
sudo puppet cert --sign clientname
```

Look at the agent console while doing this and see the handshake that is going on as the machines are joined together.

To test if an agent system can see the server, there is a command that can be used to test.

05 Introducing some Puppet basics

Before all the interesting code creation, you need to understand how Puppet works. All the configurations are held in manifest files. Manifests are just source files are what we can edit. All source files end in `.pp`

The whole point of having a Puppet setup is to ensure that the machines on the site are all the same (we can differentiate between server

types later!). To help with this aim, Puppet thoughtfully created a site-wide basic config file called `site.pp`. This is a basic file that is used to create the configurations.

To create changes on systems, a manifest is used. A manifest is a number of (or just one) text files. Within these manifests are the details that configure each part of the system that can be edited and customised. Looking at a very, very basic manifest – it is fairly clear as what it does...

```

file {'myfile':
  path    => '/tmp/myfile',
  ensure => present,
  mode    => 0640,
  content => "This could be anything.",
}
    
```

The first line is termed a resource. Resources are groups of similar things that can be configured to meet a desired standard. Examples of resources include directories, services and files. In other words, basically groups of items that share a commonality.

The bit after the file resource is what is known as the title. It can be thought of as the unique identifier. The bits that follow the identifier are properties and values. To explain it a bit better, the resource 'file' has a number of properties, such as the path and the file rights.

```

stu@puppet..ifests -- ssh
file {'myfile':
  path    => '/tmp/myfile',
  ensure => present,
  mode    => 0640,
  content => "This could be anything.",
}
~
~
~
~
~
~
~
~
~
~
    
```

```

stu@puppetmaster:/etc/puppet/manifests$ sudo puppet apply test.pp
notice: /Stage[main]/File[myfile]/ensure: created
notice: Finished catalog run in 0.16 seconds
stu@puppetmaster:/etc/puppet/manifests$
    
```

06 Testing the Puppet manifests

Manifests can be tested on the local Puppet master machine if you want to (not best practice, but will suffice for the tutorial needs). Simply save the above into a file, for example `test.pp`. Once you have done that, use the command:

```
sudo puppet apply /path/to/test.php
```

One item by itself is not very useful, so we could group together several items in one file. However, it makes more sense to split down the manifests into the jobs they do – or, to use the proper term, classes. That way you can modify the manifests to meet the requirements for multiple groups.

```

package { "apache2":
  ensure => "present"
}

service { "apache2":
  enable => "present"
}
    
```

07 Doing useful stuff with the manifest

It was mentioned before that we could do useful things with Puppet. For example, it can be made to install an application. This can be done by defining the Resource; this time the resource is 'package' and using the ensure property followed by 'ensure' to make sure it is installed or

“The whole point of having a Puppet setup is to ensure that the machines on the site are all the same”

'absent' to make sure it is not!

```
package { "apache2":
  ensure => "present"
}
```

With a simple addition, that basic start can be built up to autostart. The resource this time is 'service'. Following on from the above, add:

```
service { "apache2":
  enable => 'true'
}
```

```

class webserver {
  package { "apache2":
    ensure => "present";
  }

  package { "php5":
    ensure => "present";
  }

  service { "apache2":
    enable => 'true'
  }
}
    
```

08 Doing it cleanly with classes

Putting all these entries in one file is going to get messy, right? Also what if there are several different configurations? This is where the system can be used to differentiate. Use classes to group together bits of code that need to run, but reference it rather than putting all the code directly into site.pp.

If there was a need for a separate web server config and a database configuration, they'll have some commonalities and some differences.

So create a folder called classes under the manifest folder.

Create a new file under classes, call it `webserver.pp` and put in the following:

```

class webserver {
  package { "apache2":
    ensure => "present";
  }
  package { "php5":
    ensure => "present";
  }
  service { "apache2":
    enable => 'true'
  }
}
    
```

```

# /etc/puppet/manifests/site.pp

import "classes/*"

node default {
  include webserver
}
    
```

09 Making the class useful!

Once the class is created, it can be referenced in the site.pp file. To make it work, the classes need to be included in the latter. Go back to site.pp and modify it to include the following text:

```

# /etc/puppet/manifests/site.pp
import "classes/*"
node default {
  include webserver
}
    
```

To import the classes, we just use the import command. The 'default' means it is applied to all nodes. Notice how we use 'include webserver' and the class is called webserver? Basically, the class can be called by using 'include' suffixed by the class created that is to be referenced.

The default node is applied to all the nodes. It is possible to create nodes with special uses and work only on specific nodes. These nodes are the same layout as the default, except they have different names. Again, the include can be used to apply several configurations to all new nodes.

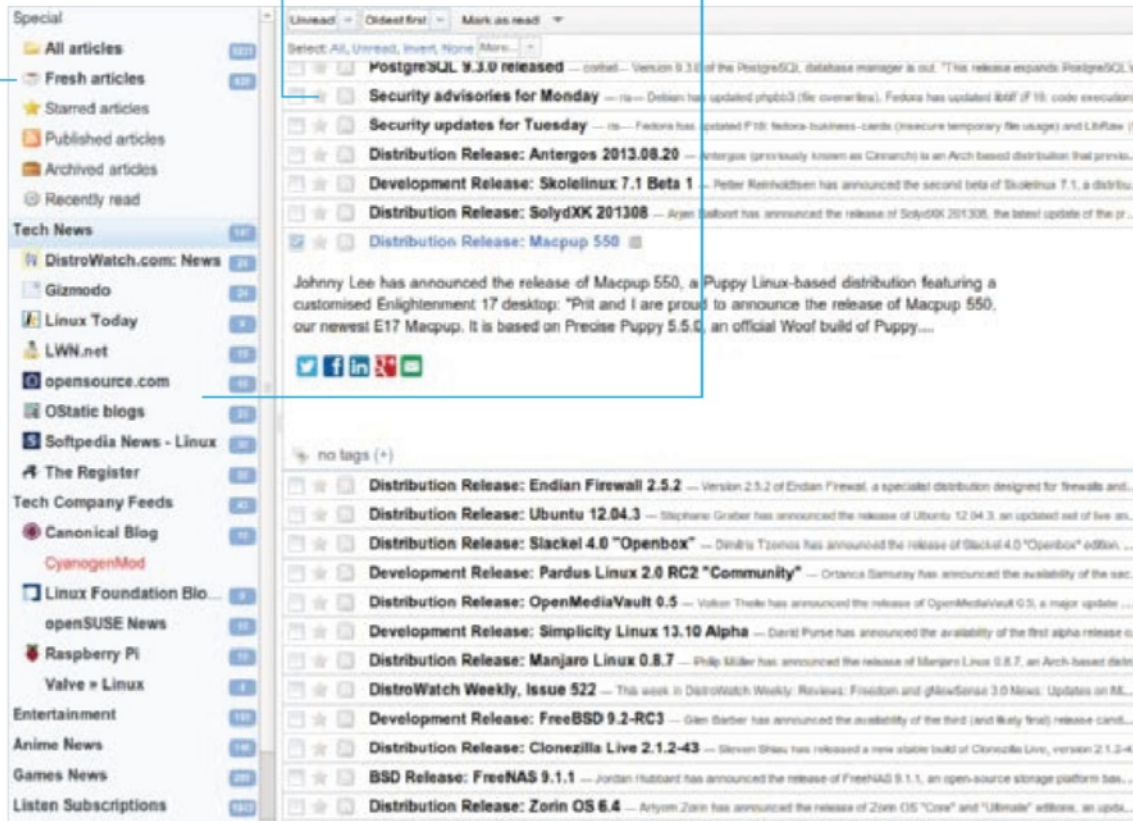
10 Assembling all the parts of Puppet

How do we group machines together and apply specifics? It's quite straightforward. Use the 'node' prefix. Again it goes into site.pp. An example of adding specific machines is:

Multiple update options to help you save server load, or allow feeds to update even when you're not looking

Highly customisable to work like Google Reader, or any other feed reader you might prefer

Recover all your own feeds from Google Reader, or make some new ones for your own feed reader



Create your own Google Reader

Use Tiny Tiny RSS to create a browser-accessible RSS feed reader located entirely on one central server

Resources

Tiny Tiny RSS:

tt-rss.org/redmine/projects/tt-rss/wiki

A server or some web space

Since Google Reader was discontinued back in 2013, users have been looking for a replacement. While some people have turned to commercial services such as Digg Reader and Feedly, the best solution is to create your own, customisable feed reader hosted entirely on your own computer or home server.

To do this, we turn to Tiny Tiny RSS, an application that has existed for a while now but has gained recent fame as a Google Reader replacement. It requires little more than a server

with a LAMP stack, or some decent web space on which you have access to the database.

Tiny Tiny RSS has a few features similar to Google Reader – keyboard shortcuts, the ability to share to different social networks, podcast recognition – and it even has an Android app so that your feeds can truly be accessed easily from anywhere with an internet connection. If you still have your old Google Reader feed download, you can also plug that into Tiny Tiny RSS.



01 Web account

On your web space or server, make sure you have a user account that can access the space or server via SSH, as we'll be setting up Tiny Tiny RSS over an SSH connection at first. Usually, you'll be able to then log in at the terminal with:

```
$ ssh username@webdomain.com
```

...by entering the relevant password.



02 MySQL database

Some web hosts have their own database creation tools, but if you're managing it yourself you'll have to create it in the command line. Create our new database with:

```
$ mysql -u root -p -h mysqlhost.mydomain.com CREATE DATABASE mydomain_ttrss
```

...replacing mysqlhost and mydomain with your relevant information.

03 MySQL user

You'll also need a user connected to the database. Most GUI tools will do this automatically; otherwise set it up with:

```
$ mysql -u root -p -h mysqlhost.mydomain.com GRANT alter,create,delete,drop,insert,update,select ON mydomain_ttrss.* TO [username] IDENTIFIED BY [password]
```

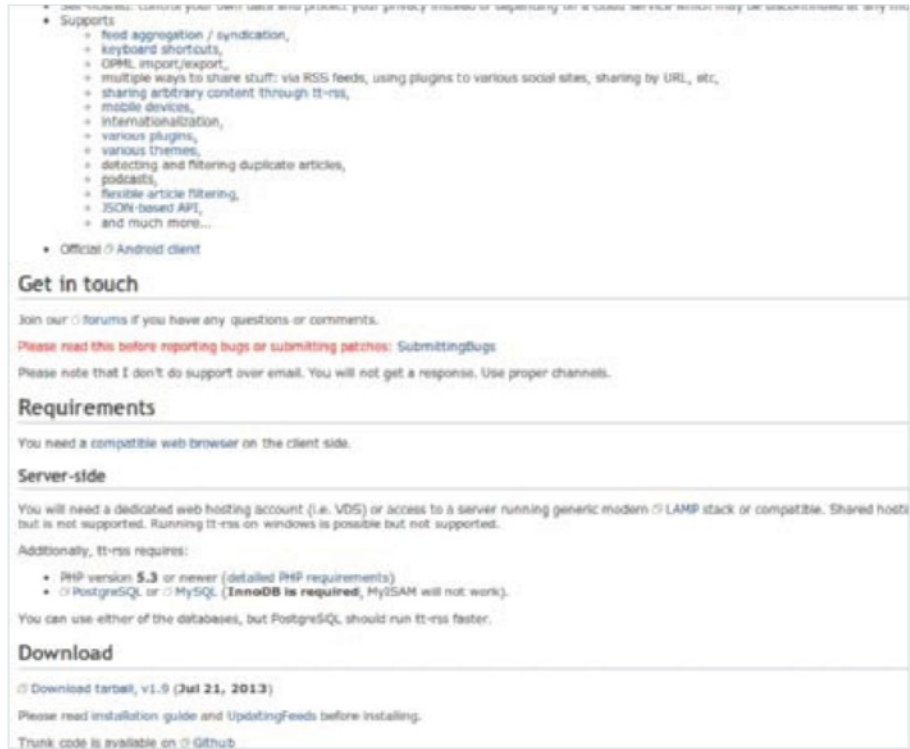
04 Get Tiny Tiny RSS

Download the latest tarball of Tiny Tiny RSS from the feed reader's website, then make sure to extract it. We'll use specific files from here to finish the database creation, then set up the rest of Tiny Tiny RSS with it.



05 Database setup

Find the MySQL sql file in the scheme



folder of files you just extracted. If you have phpMyAdmin or some other graphical MySQL interface, navigate there and import that file to the database we created. Click on Go once you've selected the file, and wait.

06 Database CLI

You can alternatively upload the scheme using SSH – cd to the folder, log in via SSH and then use this command:

```
mysql -u [username] -D mydomain_ttrss -p -h mysqlhost.mydomain.com < schema/ttrss_schema_mysql.sql
```

You'll be prompted for the user's password.

07 Copy configure

We'll now need to edit the configuration file that Tiny Tiny RSS uses to connect to the database and other services on your website, config.php-dist. Before we start messing around with it, make sure to create a copy so you can restore it and/or start again. You can find it in the top level of the directory we extracted.

08 Configuration

Rename the copied file to be config.php and then open it in a text editor. Edit the following lines with the relevant information:

```
define('DB_TYPE', "mysql");
define('DB_HOST', "mysqlhost.mydomain.com");
define('DB_USER', "[username]");
define('DB_NAME', "mydomain_ttrss");
define('DB_PASS', "[password]");
//define('DB_PORT', '');
```

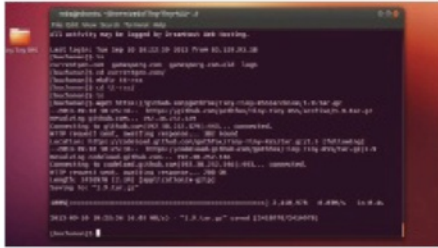


09 Set URL

Now set an URL that you'll be installing Tiny Tiny RSS to. Scroll down to and locate:

```
define('SELF_URL_PATH', 'http://example.org/tt-rss/');
```

Change example.org to your domain name, and you can either keep tt-rss or choose something else – this is where you're going to be installing Tiny Tiny RSS.



10 Downloading After you've saved the file, go back in the terminal and log into your web space. Create the tt-rss folder with:

```
$ mkdir tt-rss
```

Move into the directory, then download the latest version of Tiny Tiny RSS with:

```
$ wget https://github.com/gothfox/Tiny-Tiny-RSS/archive/1.9.tar.gz
```

11 Extracting Extract the files into the directory using the following:

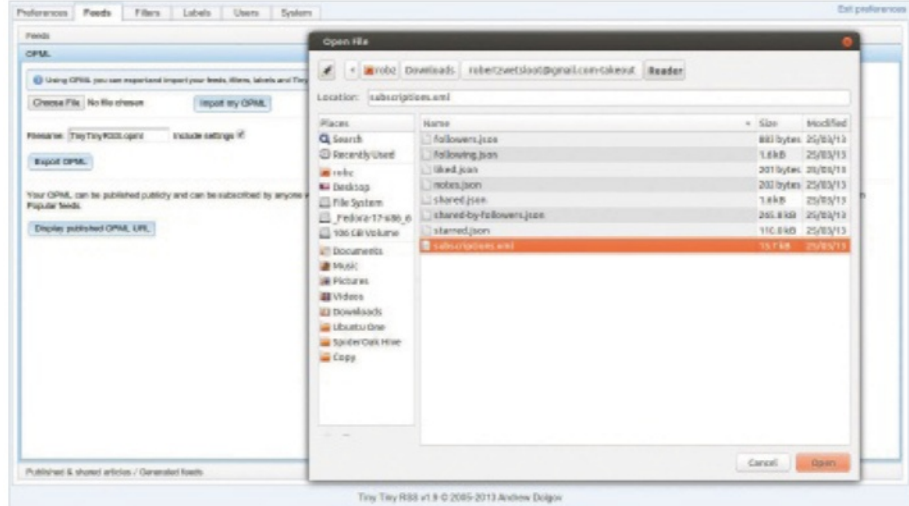
```
$ tar -zxvf 1.9.tar.gz
```

Make sure that all the files extracted are moved from the folder created into the tt-rss parent folder, so that when we're navigating there, it opens the right files.

12 Updating Move the config file we created to your web space via FTP or a download method, or just edit the config file the same way we did earlier. You can also use a graphical FTP program to move the files to the space without having to do anything in the terminal.



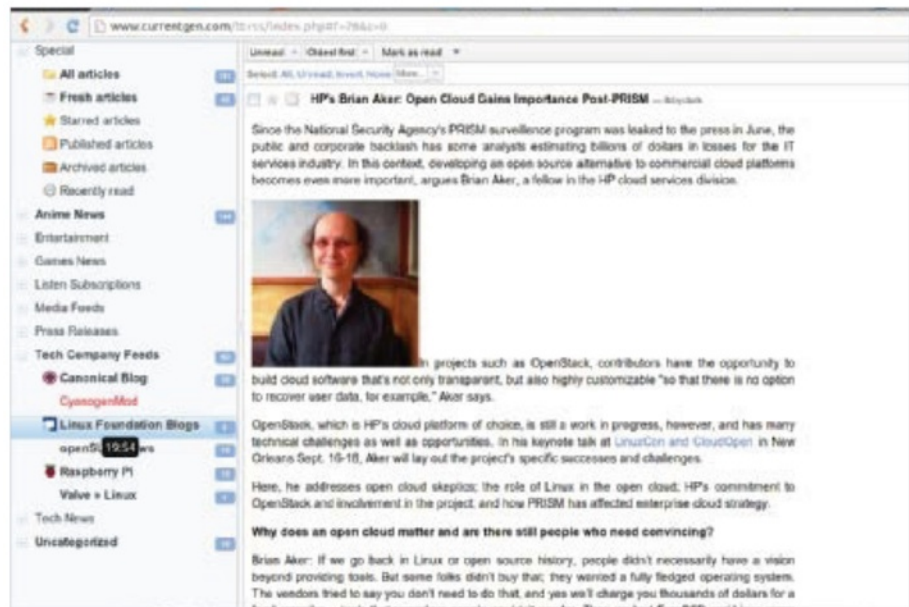
13 Login Now we can log in. Go to the URL we specified and you'll see the login screen – for now, this will be admin and password. Once logged in, go to Actions>Preferences>Personal Data/Authentication and change the password.



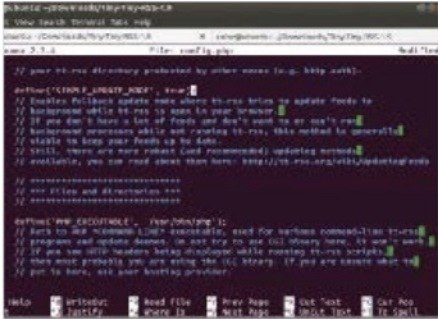
14 Import feeds If you still have the Takeout data from Google Reader, you can now import this into Tiny Tiny RSS. Go back to the Actions menu and click Feeds, then OPML. Click 'Choose file', navigate to your subscriptions.xml and then click Import. It will keep your folder structure if you had one.

15 Reading Be aware that Tiny Tiny RSS won't work

like Google Reader right away, but you can exit the preferences to see how the rough layout is. Don't be concerned if your feeds don't show anything in them – we will be fixing that in a bit. For now, if you double-click on the individual feeds in your folders, you can load up some of the latest entries. You can also start editing the way feed items are displayed, with oldest or newest first and whether or not to show read items.



“Tiny Tiny RSS won't work like Google Reader right away”



16 Easy updating

There are two ways to get Tiny Tiny RSS to update – the easiest is to go back into the config.php file and locate the 'SIMPLE_UPDATE_MODE' line. Change it to true and it will automatically update your feeds while you're browsing your feed reader

17 Better updating

If you're running Tiny Tiny RSS on your own server, you can have the update daemons run in the background so that it's always up to date when you log in. For single processes, access the PHP CLI and use:

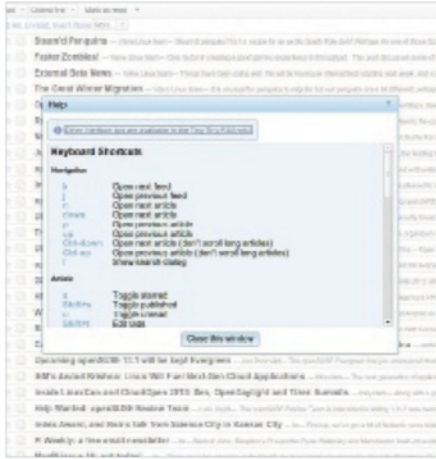
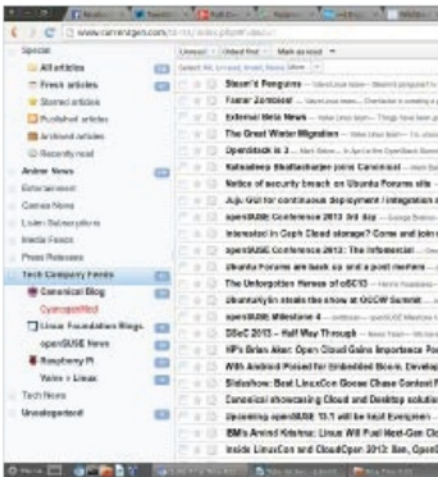
```
php ./update.php daemon
```

For multi-process use:

```
php ./update_daemon2.php
```

18 View

By default, all the articles will be in the expanded view – a nightmare for properly navigating your feeds. To change it so that you can click and navigate between feed items, go to Preferences and uncheck 'Automatically expand articles in combined mode'.



19 Keyboard shortcuts

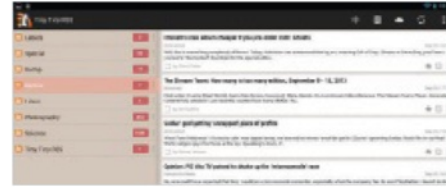
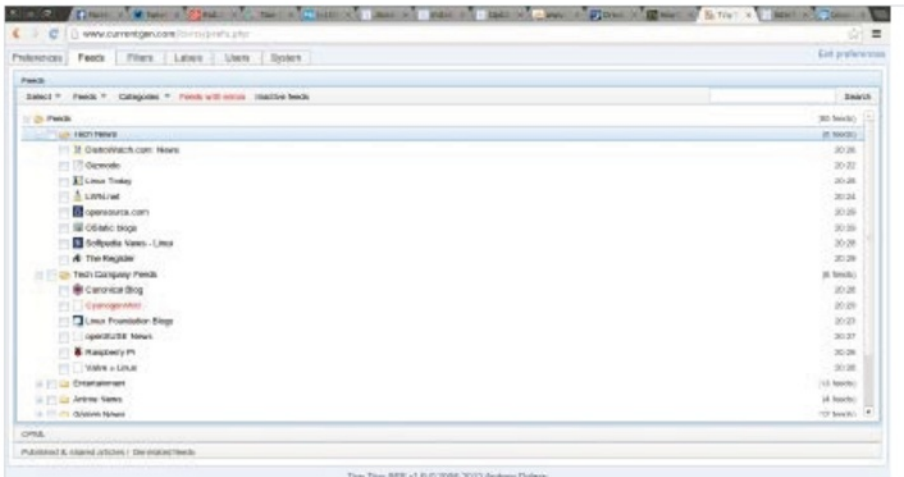
The keyboard shortcuts are slightly different in Tiny Tiny RSS. For going between articles, arrow-up and down or N and P are used – J and K are now used to go between feeds. You can view the full list by going to Actions, then 'Keyboard shortcuts help'.

20 Add feeds

Go to Actions>Subscribe to Feed and paste an URL into the field. You can then assign it to one of your categories/folders, then hit Subscribe. If you want to move it to another category, go to Preferences>Feeds and then right-click on the necessary subscription

21 Rearrange categories

It may seem from the main interface that you can't rearrange your folders to an order you prefer. However, go back to Preferences>Feeds and you can drag and drop them to your requirements. You can even move feeds between categories this way.



22 Android

If you used to read your feeds on the go with Google Reader on Android, you'll be glad to know there's an Android app for Tiny Tiny RSS. It will cost just over £1 to use it after seven days, but it's a great way to keep up with your feeds on mobile, and easily logs into your new RSS reader.



23 Updating

Like all good software, Tiny Tiny RSS is undergoing upgrades and updates all the time. You can update the software without reinstalling, though – simply go to Preferences and look for the Update Tiny Tiny RSS tab. It will inform you of any available updates.

24 Keep tweaking

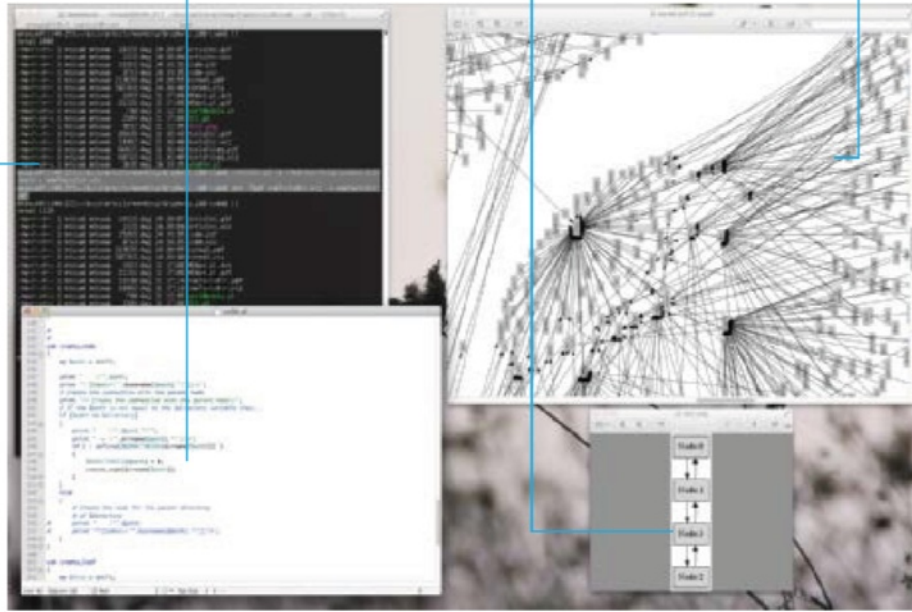
There's a lot of other little tweaks you can do with Tiny Tiny RSS, such as the update interval, the length of time to store articles, what's considered a fresh item, etc. Go through the preferences and make Tiny Tiny RSS the best reader for your own needs.

Running the visDir.pl Perl script and creating a PDF file using dot

Part of the visDir.pl Perl script

A simple Graphviz graph

A small part of the Linux kernel directory structure!



Visualise directory structures with Graphviz

Make large directory structures practical with this open source visualisation package

Resources

Graphviz: www.graphviz.org

A text editor

Perl: www.perl.org

Perl Graphviz module: search.cpan.org/~lbocard/GraphViz-2.02/lib/GraphViz.pm

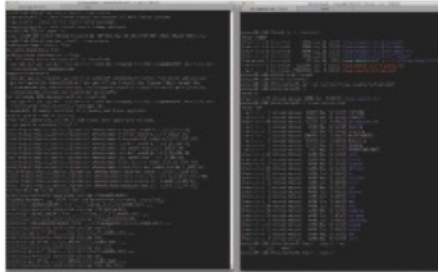
Graphviz output formats: www.graphviz.org/doc/info/output.html

Graphviz is a collection of tools for manipulating graph structures and generating graph layouts. It supports both directed and undirected graphs and offers graphical and command-line tools – we’ll be using the latter.

Graphviz contains many programs and libraries. The dot program is a utility for drawing directed graphs. It accepts input in the dot language. The dot language can define three kinds of objects: graphs, nodes and edges. Neato is a program for drawing undirected graphs, which are commonly used for telecoms and computer programming tasks. The circo utility is used for creating circular layouts of graphs, while fdp generates undirected graphs. The sfdp program is a utility for constructing large undirected graphs. The twopi program

is a utility for drawing graphs using a circular layout. One node is chosen as the centre, and the other nodes are placed around the centre in a circular pattern. If a node is connected to the centre node, it is placed at distance 1. If a node is connected to a node directly connected to the centre node, it is placed at distance 2 and so on.

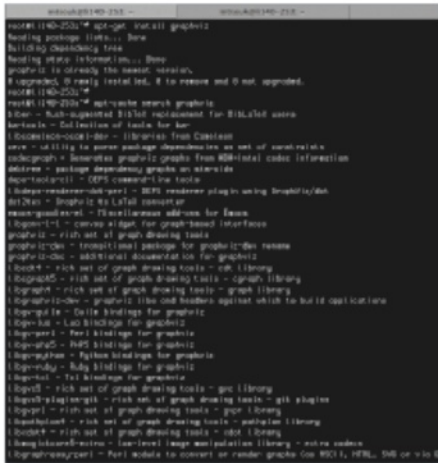
Graphviz also provides three graphical programs named dotty, tcldot and lefty: lefty is a graphical editor for technical pictures; dotty is a customisable interface for the X Window System written in lefty; tcldot is a customisable graphical interface written in Tcl 7. There are also two drawing libraries called libgraph and libgraph. Their existence means that an application can use Graphviz as a library rather than as a software tool.



01 Why Graphviz?

Visualising large directory structures such as the Linux kernel can be really practical. The Linux kernel root directory contains over 2,000 other directories and 37,000 files that would otherwise be very difficult to picture. The output of our Perl script can optionally show the included files as well as their sizes. Also, the Graphviz knowledge you'll get by visualising directories can be used for visualising networks, traceroute paths, function calls etc. And there are plenty of other benefits.

Note: For huge directory structures such as the Linux kernel, it is better not to visualise all at once but to split into smaller parts.



02 Installing and running Graphviz

Your Linux distribution probably includes a ready-to-install Graphviz package that you can use. For a Debian 7 system, you just have to run the following command to download and install Graphviz:

```
# apt-get install graphviz
```

After installing Graphviz, try to compile the following Graphviz code:

```
digraph G
{
```



```
"Hello world!";
}
```

Use the following command for the compilation:

```
$ dot -Tps hw.dot -o hw.ps
```

The aforementioned command will produce a PostScript file called hw.ps that you can view. The word digraph means that a directed graph is going to be created. For creating an undirected graph, the word graph should have been used instead. For such a simplistic example, however, it does not make any difference if the graph is either directed or undirected.

Although the PostScript format used to be very popular, it is recommended to use the PDF format because it is faster to render and display. Additionally, PDF files can be zoomed in more before losing their clarity.

03 The Perl Graphviz module

Many programming languages, including Python, Ruby, C++ and Perl, provide their own interface for creating Graphviz files. The Perl module is presented as an alternative way of generating Graphviz code.

The important thing to remember when using the Graphviz module is that if you want to get the output as a PNG file, the last line of your program should be:

```
print $graph->as_png;
```

Similarly, if you want to get the output as plain text, you should use the following line instead:

```
print $graph->as_text;
```



04 Basic Graphviz information

A graph $G(V,E)$ is a finite, non-empty set of vertices V (or nodes) and a set of edges E . A graph contains nodes and edges, each of them having attributes.

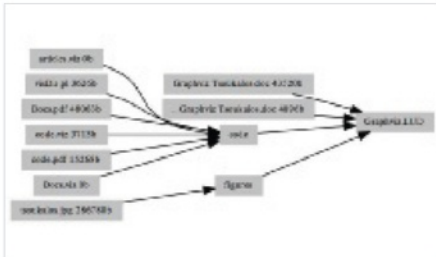
Graphviz has its own dialect that you will have to learn. The language may be simple and elegant but it is also very powerful. The good thing about Graphviz is that you can write its code using a simple plain text editor – a wonderful side effect of it is that you can easily write scripts that generate Graphviz code.

By reading some Graphviz code, you will soon realise that lines beginning with $\#$ or $//$ are considered comments.

Node Attributes		
Name	Explanation	Allowed Values
shape	The shape of the node	ellipse, diamond, box, etc
height	The height in inches	a number
width	The width in inches	a number
label	The name of the node	alphanumeric
fontsize	The size of the font	a number
fontname	The name of the font	Courier, Helvetica, Times
fontcolor	The color of the font	white, black, blue, etc
style	The style name	bold, dotted, filled, etc.
color	The color of the node shape	white, black, etc.
pos	The coordinates of the position	

Edge Attributes		
Name	Explanation	Allowed Values
label	The label of the edge	alphanumeric
fontsize	The size of the font	
fontname	The name of the font	
fontcolor	The color of the font	
style	The style name	bold, dotted, filled, etc.
color	The color of the edge	white, black, blue, etc.
len	The length of the edge	
dir	The direction of the edge	forward, back, both or none
decorate	Draws a line that connects labels with their edges	0 or 1
id	Optional value to denote different edges	alphanumeric

“Experiment to find the suitable tool and parameters for the job”



05 A simple Graphviz example

The following Graphviz code draws a simple directory structure that includes files. It also displays the size of a file:

```
digraph Widget
{
    size="16,6";
    nodesep=0.05;
    rankdir = LR;
    rotate = 90;
    edge[len=5];
    node[style=filled, shape=record,
fontsize=8];
    node[height=0.20, width=0.20,
color=gray];
    "Graphviz Tsoukalos.doc
doc"[label="Graphviz Tsoukalos.doc
43520b"];
    "Graphviz Tsoukalos.doc" -> "/home/
mtsouk/docs/article/working/Graphviz.LUD";
    "/home/mtsouk/docs/article/working/
Graphviz.LUD"[label="Graphviz.LUD"];
    "_Graphviz Tsoukalos.doc"[label="._
Graphviz Tsoukalos.doc 4096b"];
    "_Graphviz Tsoukalos.doc" -> "/home/
mtsouk/docs/article/working/Graphviz.LUD";
    "/home/mtsouk/docs/article/working/
Graphviz.LUD/code"[label="code"];
    "/home/mtsouk/docs/article/working/
Graphviz.LUD/code" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD";
```

```
"articles.viz"[label="articles.viz
0b"];
"articles.viz" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"visDir.pl"[label="visDir.pl 3626b"];
"visDir.pl" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"Docs.pdf"[label="Docs.pdf 48063b"];
"Docs.pdf" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"code.viz"[label="code.viz 3713b"];
"code.viz" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"code.pdf"[label="code.pdf 15268b"];
"code.pdf" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"Docs.viz"[label="Docs.viz 0b"];
"Docs.viz" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"/home/mtsouk/docs/article/working/
Graphviz.LUD/figures"[label="figures"];
"/home/mtsouk/docs/article/working/
Graphviz.LUD/figures" -> "/home/mtsouk/
docs/article/working/Graphviz.LUD";
"tsoukalos.jpg"[label="tsoukalos.jpg
286780b"];
"tsoukalos.jpg" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/figures";
}
```

The code (articles.viz) can be compiled using the following command:

```
$ dot -Tpdf articles.viz -o articles.pdf
```

The presented visDir.pl Perl script generates similar Graphviz code.

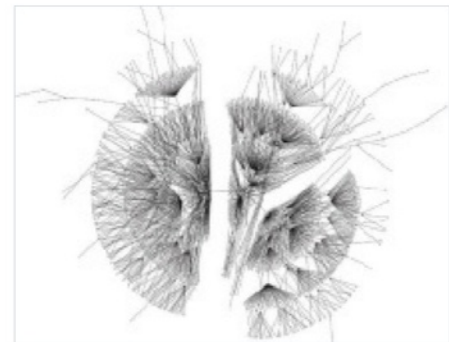
06 A complicated Graphviz example

You can visualise the Linux kernel directory structure with the help of the visDir.pl script using the following two commands:

```
$ ./visDir.pl -d ~/kernel/linux-source-3.2
```

```
> kernel.viz
$ neato -Tpdf kernel.viz -o kernel.pdf
```

As you already know, the Graphviz suite contains many tools for creating graphs. For graphs with a large number of nodes, you should experiment to find the suitable tool and parameters for the job. The output of the dot tool for the Linux kernel directory structure is not as pretty as the graph created using the neato tool.



07 The Perl script

The Perl script, called visDir.pl, requires one command-line option and one argument. The argument is the path of the directory that is going to be visualised. The command-line option must be -d (for including directories only) or -f (for also including files). If none of them is found, the script prints an explanatory message and stops execution. Please note that the directory argument must not contain a '/' at the end; so the following command will not work properly:

```
$ ./program_name.pl /usr/
```

```
35 die <<Thonot unless $ARG;
36 usage:
37 $0 [ -f | -d ] directory
38 Thonot:
39
40 if { $ARG =~ /-Z/ }
41 {
42     die <<Thonot:
43     usage info:
44     Please use exactly 2 arguments!
45     Use -f if you want to include files
46     Use -d if you want to include directories ONLY
47 Thonot:
48 }
49
50 # Get the file name
51 ($files,$directory) = @ARGV;
52
53 # print $files, "\n";
54
55 # Check the $files variable
56 if { $files eq "-d" }
57 {
58     $files = 1;
59 }
60 elsif { $files eq "-f" }
61 {
62     $files = 0;
63 }
64 else
65 {
66     die <<Thonot:
67     Use -f if you want to include files
68     Use -d if you want to include directories ONLY
69 Thonot:
70 }
```

“The Linux kernel root directory contains over 37,000 files”

```

72 print <<START;
73 digraph Widget
74 {
75     size="16,6";
76     nodesep=0.05;
77     rankdir = LR;
78     rotate = 90;
79     edge[pen=5];
80     node[style=Filled, shape=record, fontsize=8];
81     node[height=0.20, width=0.20, color=gray];
82
83 START

```

08 Explaining the Perl script – part 1

The first thing that the Perl script prints is the common attributes of all the nodes. If you want to make global changes to the output, you should look at this part of the script.

Depending on the total number of nodes of your own graph, you may need to experiment with the given part of the script to beautify the output. The 'rankdir = LR' command dictates that graph nodes must be drawn from left to right. The 'rotate = 90' command draws the output in landscape mode.

```

105 sub create_graphviz
106 {
107     # print $L;
108     # print "M";
109     # SNAIP, ... .DS_Store and All .png files
110     # You can change it to include the files you want
111     if ( $L =~ /^\.\/\./ || $L =~ /\.DS_Store$/ || $L =~ /\.png$/ )
112     {
113         # do nothing!
114     }
115     else
116     {
117         # If it is a directory, then ...
118         if ( -d $files::find::name )
119         {
120             # duplicates can only exist in directories.
121             # We must take care of it.
122             if ( ! defined(SUBDIRECTORIES{$files::find::name}) )
123             {
124                 SUBDIRECTORIES{$files::find::name} = 0;
125                 create_node($files::find::name);
126             }
127             # If you do not want to display files
128             # skip $files::find::name
129             # do nothing!
130             # If it is a file, then ...
131             else
132             {
133                 create_leaf($files::find::name);
134             }
135         }
136     }
137 }

```

```

161 sub create_leaf
162 {
163     my $file = shift;
164     my $size = 0;
165
166     # It is always a good idea to check twice!
167     if ( -f $file )
168     {
169         # This finds the size of the file in bytes
170         $size = -s $file;
171     }
172     # add the byte symbol at the end of the byte number
173     $size = "b";
174
175     # create the file node
176     print "    [" . $file . " " . $size . "]\n";
177     print "    [" . $file . "]\n";
178     print "    [" . $file . "]\n";
179     if ( ! defined(SUBDIRECTORIES{$files::find::name}) )
180     {
181         SUBDIRECTORIES{$files::find::name} = 0;
182         create_node($files::find::name);
183     }
184 }
185 }
186

```

09 Explaining the Perl script – part 2

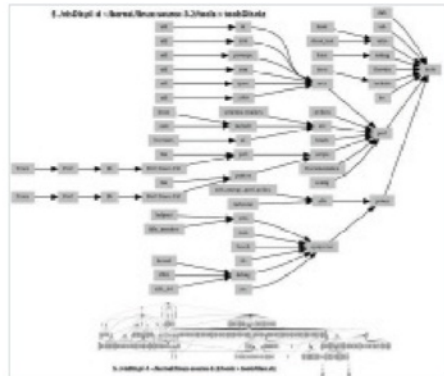
The create_graphviz function does most of the work. First, it excludes files or

directories that match given patterns, using the following command:

```
if ( $L =~ /^\.\/\./ || $L =~ /\.DS_Store$/ || $L =~ /\.png$/ )
```

Second, every time a directory is found, it calls the create_node function to construct a new node for it.

Third, if it has to process a file, it calls the create_leaf function (files cannot include other files so they are considered the leaves of a graph). The create_leaf function not only produces a node for the given file, but also finds out the file size and includes it in the output.



10 Explaining the Perl script – part 3

The following Perl code checks if the -d option is given in order to exclude all files from the output:

```
if ( $nofiles eq "-d" )
{
    $nofiles = 1;
}

```

Should you want to exclude Word documents (their file extension is either doc or docx) from the output but include PNG files, you should use the following code:

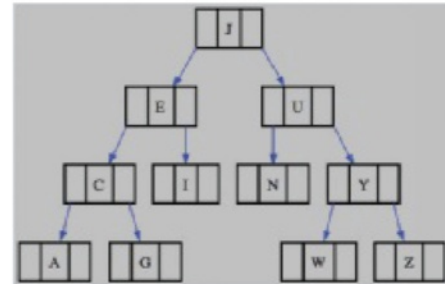
```
if ( $L =~ /^\.\/\./ || $L =~ /\.DS_Store$/ || $L =~ /\.doc$/ || $L =~ /\.docx$/ )
```

Note: The first regular expression (/^\.\/\./) excludes '.' and '..' directories from the output. '.DS_Store' files are also excluded from the output as they are Mac OS X specific and are used for storing custom directory attributes.

As far as the Linux kernel source is concerned, it would be a good idea to exclude Makefile and Kconfig files as well as object files from the output in order to simplify it. Excluding Makefile

and Kconfig files is as easy as using the following line of code:

```
if ( $L =~ /^\.\/\./ || $L =~ /\.Makefile$/ || $L =~ /\.Kconfig$/ )
```



11 More about Graphviz

Graphviz can also be used for creating database relation diagrams between tables, UML diagrams, website maps, source code diagrams, computer network diagrams, binary trees, hash tables and knowledge representation diagrams. The good thing is that Graphviz algorithmically arranges the graph nodes so that the output is both practical and pleasing!

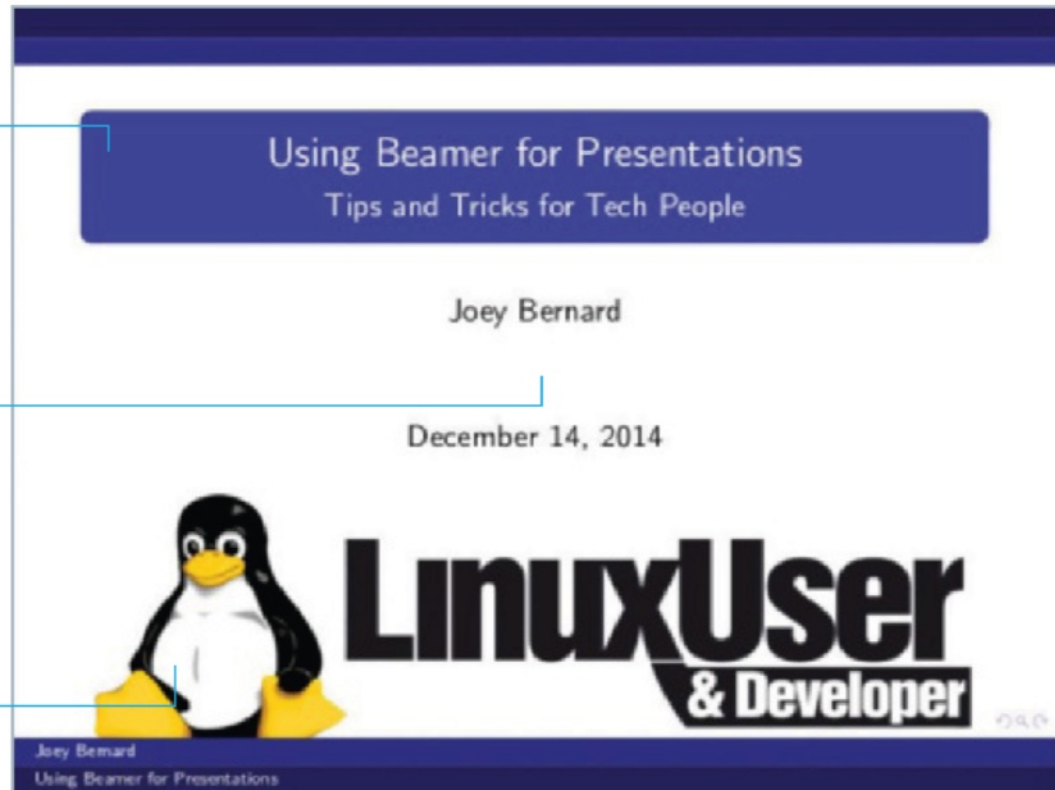
```

1 digraph G
2 {
3     graph TD
4     node0["node:0"] --> node1["node:1"]
5     node0 --> node2["node:2"]
6     node1 --> node3["node:3"]
7     node1 --> node4["node:4"]
8     node2 --> node5["node:5"]
9     node2 --> node6["node:6"]
10    node3 --> node7["node:7"]
11    node3 --> node8["node:8"]
12    node4 --> node9["node:9"]
13    node4 --> node10["node:10"]
14    node5 --> node11["node:11"]
15    node5 --> node12["node:12"]
16    node6 --> node13["node:13"]
17    node6 --> node14["node:14"]
18    node7 --> node15["node:15"]
19    node7 --> node16["node:16"]
20    node8 --> node17["node:17"]
21    node8 --> node18["node:18"]
22    node9 --> node19["node:19"]
23    node9 --> node20["node:20"]
24    node10 --> node21["node:21"]
25    node10 --> node22["node:22"]
26    node11 --> node23["node:23"]
27    node11 --> node24["node:24"]
28    node12 --> node25["node:25"]
29    node12 --> node26["node:26"]
30    node13 --> node27["node:27"]
31    node13 --> node28["node:28"]
32    node14 --> node29["node:29"]
33    node14 --> node30["node:30"]
34    node15 --> node31["node:31"]
35    node15 --> node32["node:32"]
36    node16 --> node33["node:33"]
37    node16 --> node34["node:34"]
38    node17 --> node35["node:35"]
39    node17 --> node36["node:36"]
40    node18 --> node37["node:37"]
41    node18 --> node38["node:38"]
42    node19 --> node39["node:39"]
43    node19 --> node40["node:40"]
44    node20 --> node41["node:41"]
45    node20 --> node42["node:42"]
46    node21 --> node43["node:43"]
47    node21 --> node44["node:44"]
48    node22 --> node45["node:45"]
49    node22 --> node46["node:46"]
50    node23 --> node47["node:47"]
51    node23 --> node48["node:48"]
52    node24 --> node49["node:49"]
53    node24 --> node50["node:50"]
54    node25 --> node51["node:51"]
55    node25 --> node52["node:52"]
56    node26 --> node53["node:53"]
57    node26 --> node54["node:54"]
58    node27 --> node55["node:55"]
59    node27 --> node56["node:56"]
60    node28 --> node57["node:57"]
61    node28 --> node58["node:58"]
62    node29 --> node59["node:59"]
63    node29 --> node60["node:60"]
64    node30 --> node61["node:61"]
65    node30 --> node62["node:62"]
66    node31 --> node63["node:63"]
67    node31 --> node64["node:64"]
68    node32 --> node65["node:65"]
69    node32 --> node66["node:66"]
70    node33 --> node67["node:67"]
71    node33 --> node68["node:68"]
72    node34 --> node69["node:69"]
73    node34 --> node70["node:70"]
74    node35 --> node71["node:71"]
75    node35 --> node72["node:72"]
76    node36 --> node73["node:73"]
77    node36 --> node74["node:74"]
78    node37 --> node75["node:75"]
79    node37 --> node76["node:76"]
80    node38 --> node77["node:77"]
81    node38 --> node78["node:78"]
82    node39 --> node79["node:79"]
83    node39 --> node80["node:80"]
84    node40 --> node81["node:81"]
85    node40 --> node82["node:82"]
86    node41 --> node83["node:83"]
87    node41 --> node84["node:84"]
88    node42 --> node85["node:85"]
89    node42 --> node86["node:86"]
90    node43 --> node87["node:87"]
91    node43 --> node88["node:88"]
92    node44 --> node89["node:89"]
93    node44 --> node90["node:90"]
94    node45 --> node91["node:91"]
95    node45 --> node92["node:92"]
96    node46 --> node93["node:93"]
97    node46 --> node94["node:94"]
98    node47 --> node95["node:95"]
99    node47 --> node96["node:96"]
100   node48 --> node97["node:97"]
101   node48 --> node98["node:98"]
102   node49 --> node99["node:99"]
103   node49 --> node100["node:100"]
104   node50 --> node101["node:101"]
105   node50 --> node102["node:102"]
106   node51 --> node103["node:103"]
107   node51 --> node104["node:104"]
108   node52 --> node105["node:105"]
109   node52 --> node106["node:106"]
110   node53 --> node107["node:107"]
111   node53 --> node108["node:108"]
112   node54 --> node109["node:109"]
113   node54 --> node110["node:110"]
114   node55 --> node111["node:111"]
115   node55 --> node112["node:112"]
116   node56 --> node113["node:113"]
117   node56 --> node114["node:114"]
118   node57 --> node115["node:115"]
119   node57 --> node116["node:116"]
120   node58 --> node117["node:117"]
121   node58 --> node118["node:118"]
122   node59 --> node119["node:119"]
123   node59 --> node120["node:120"]
124   node60 --> node121["node:121"]
125   node60 --> node122["node:122"]
126   node61 --> node123["node:123"]
127   node61 --> node124["node:124"]
128   node62 --> node125["node:125"]
129   node62 --> node126["node:126"]
130   node63 --> node127["node:127"]
131   node63 --> node128["node:128"]
132   node64 --> node129["node:129"]
133   node64 --> node130["node:130"]
134   node65 --> node131["node:131"]
135   node65 --> node132["node:132"]
136   node66 --> node133["node:133"]
137   node66 --> node134["node:134"]
138   node67 --> node135["node:135"]
139   node67 --> node136["node:136"]
140   node68 --> node137["node:137"]
141   node68 --> node138["node:138"]
142   node69 --> node139["node:139"]
143   node69 --> node140["node:140"]
144   node70 --> node141["node:141"]
145   node70 --> node142["node:142"]
146   node71 --> node143["node:143"]
147   node71 --> node144["node:144"]
148   node72 --> node145["node:145"]
149   node72 --> node146["node:146"]
150   node73 --> node147["node:147"]
151   node73 --> node148["node:148"]
152   node74 --> node149["node:149"]
153   node74 --> node150["node:150"]
154   node75 --> node151["node:151"]
155   node75 --> node152["node:152"]
156   node76 --> node153["node:153"]
157   node76 --> node154["node:154"]
158   node77 --> node155["node:155"]
159   node77 --> node156["node:156"]
160   node78 --> node157["node:157"]
161   node78 --> node158["node:158"]
162   node79 --> node159["node:159"]
163   node79 --> node160["node:160"]
164   node80 --> node161["node:161"]
165   node80 --> node162["node:162"]
166   node81 --> node163["node:163"]
167   node81 --> node164["node:164"]
168   node82 --> node165["node:165"]
169   node82 --> node166["node:166"]
170   node83 --> node167["node:167"]
171   node83 --> node168["node:168"]
172   node84 --> node169["node:169"]
173   node84 --> node170["node:170"]
174   node85 --> node171["node:171"]
175   node85 --> node172["node:172"]
176   node86 --> node173["node:173"]
177   node86 --> node174["node:174"]
178   node87 --> node175["node:175"]
179   node87 --> node176["node:176"]
180   node88 --> node177["node:177"]
181   node88 --> node178["node:178"]
182   node89 --> node179["node:179"]
183   node89 --> node180["node:180"]
184   node90 --> node181["node:181"]
185   node90 --> node182["node:182"]
186   node91 --> node183["node:183"]
187   node91 --> node184["node:184"]
188   node92 --> node185["node:185"]
189   node92 --> node186["node:186"]
190   node93 --> node187["node:187"]
191   node93 --> node188["node:188"]
192   node94 --> node189["node:189"]
193   node94 --> node190["node:190"]
194   node95 --> node191["node:191"]
195   node95 --> node192["node:192"]
196   node96 --> node193["node:193"]
197   node96 --> node194["node:194"]
198   node97 --> node195["node:195"]
199   node97 --> node196["node:196"]
200   node98 --> node197["node:197"]
201   node98 --> node198["node:198"]
202   node99 --> node199["node:199"]
203   node99 --> node200["node:200"]
204   node100 --> node201["node:201"]
205   node100 --> node202["node:202"]
206   node101 --> node203["node:203"]
207   node101 --> node204["node:204"]
208   node102 --> node205["node:205"]
209   node102 --> node206["node:206"]
210   node103 --> node207["node:207"]
211   node103 --> node208["node:208"]
212   node104 --> node209["node:209"]
213   node104 --> node210["node:210"]
214   node105 --> node211["node:211"]
215   node105 --> node212["node:212"]
216   node106 --> node213["node:213"]
217   node106 --> node214["node:214"]
218   node107 --> node215["node:215"]
219   node107 --> node216["node:216"]
220   node108 --> node217["node:217"]
221   node108 --> node218["node:218"]
222   node109 --> node219["node:219"]
223   node109 --> node220["node:220"]
224   node110 --> node221["node:221"]
225   node110 --> node222["node:222"]
226   node111 --> node223["node:223"]
227   node111 --> node224["node:224"]
228   node112 --> node225["node:225"]
229   node112 --> node226["node:226"]
230   node113 --> node227["node:227"]
231   node113 --> node228["node:228"]
232   node114 --> node229["node:229"]
233   node114 --> node230["node:230"]
234   node115 --> node231["node:231"]
235   node115 --> node232["node:232"]
236   node116 --> node233["node:233"]
237   node116 --> node234["node:234"]
238   node117 --> node235["node:235"]
239   node117 --> node236["node:236"]
240   node118 --> node237["node:237"]
241   node118 --> node238["node:238"]
242   node119 --> node239["node:239"]
243   node119 --> node240["node:240"]
244   node120 --> node241["node:241"]
245   node120 --> node242["node:242"]
246   node121 --> node243["node:243"]
247   node121 --> node244["node:244"]
248   node122 --> node245["node:245"]
249   node122 --> node246["node:246"]
250   node123 --> node247["node:247"]
251   node123 --> node248["node:248"]
252   node124 --> node249["node:249"]
253   node124 --> node250["node:250"]
254   node125 --> node251["node:251"]
255   node125 --> node252["node:252"]
256   node126 --> node253["node:253"]
257   node126 --> node254["node:254"]
258   node127 --> node255["node:255"]
259   node127 --> node256["node:256"]
260   node128 --> node257["node:257"]
261   node128 --> node258["node:258"]
262   node129 --> node259["node:259"]
263   node129 --> node260["node:260"]
264   node130 --> node261["node:261"]
265   node130 --> node262["node:262"]
266   node131 --> node263["node:263"]
267   node131 --> node264["node:264"]
268   node132 --> node265["node:265"]
269   node132 --> node266["node:266"]
270   node133 --> node267["node:267"]
271   node133 --> node268["node:268"]
272   node134 --> node269["node:269"]
273   node134 --> node270["node:270"]
274   node135 --> node271["node:271"]
275   node135 --> node272["node:272"]
276   node136 --> node273["node:273"]
277   node136 --> node274["node:274"]
278   node137 --> node275["node:275"]
279   node137 --> node276["node:276"]
280   node138 --> node277["node:277"]
281   node138 --> node278["node:278"]
282   node139 --> node279["node:279"]
283   node139 --> node280["node:280"]
284   node140 --> node281["node:281"]
285   node140 --> node282["node:282"]
286   node141 --> node283["node:283"]
287   node141 --> node284["node:284"]
288   node142 --> node285["node:285"]
289   node142 --> node286["node:286"]
290   node143 --> node287["node:287"]
291   node143 --> node288["node:288"]
292   node144 --> node289["node:289"]
293   node144 --> node290["node:290"]
294   node145 --> node291["node:291"]
295   node145 --> node292["node:292"]
296   node146 --> node293["node:293"]
297   node146 --> node294["node:294"]
298   node147 --> node295["node:295"]
299   node147 --> node296["node:296"]
300   node148 --> node297["node:297"]
301   node148 --> node298["node:298"]
302   node149 --> node299["node:299"]
303   node149 --> node300["node:300"]
304   node150 --> node301["node:301"]
305   node150 --> node302["node:302"]
306   node151 --> node303["node:303"]
307   node151 --> node304["node:304"]
308   node152 --> node305["node:305"]
309   node152 --> node306["node:306"]
310   node153 --> node307["node:307"]
311   node153 --> node308["node:308"]
312   node154 --> node309["node:309"]
313   node154 --> node310["node:310"]
314   node155 --> node311["node:311"]
315   node155 --> node312["node:312"]
316   node156 --> node313["node:313"]
317   node156 --> node314["node:314"]
318   node157 --> node315["node:315"]
319   node157 --> node316["node:316"]
320   node158 --> node317["node:317"]
321   node158 --> node318["node:318"]
322   node159 --> node319["node:319"]
323   node159 --> node320["node:320"]
324   node160 --> node321["node:321"]
325   node160 --> node322["node:322"]
326   node161 --> node323["node:323"]
327   node161 --> node324["node:324"]
328   node162 --> node325["node:325"]
329   node162 --> node326["node:326"]
330   node163 --> node327["node:327"]
331   node163 --> node328["node:328"]
332   node164 --> node329["node:329"]
333   node164 --> node330["node:330"]
334   node165 --> node331["node:331"]
335   node165 --> node332["node:332"]
336   node166 --> node333["node:333"]
337   node166 --> node334["node:334"]
338   node167 --> node335["node:335"]
339   node167 --> node336["node:336"]
340   node168 --> node337["node:337"]
341   node168 --> node338["node:338"]
342   node169 --> node339["node:339"]
343   node169 --> node340["node:340"]
344   node170 --> node341["node:341"]
345   node170 --> node342["node:342"]
346   node171 --> node343["node:343"]
347   node171 --> node344["node:344"]
348   node172 --> node345["node:345"]
349   node172 --> node346["node:346"]
350   node173 --> node347["node:347"]
351   node173 --> node348["node:348"]
352   node174 --> node349["node:349"]
353   node174 --> node350["node:350"]
354   node175 --> node351["node:351"]
355   node175 --> node352["node:352"]
356   node176 --> node353["node:353"]
357   node176 --> node354["node:354"]
358   node177 --> node355["node:355"]
359   node177 --> node356["node:356"]
360   node178 --> node357["node:357"]
361   node178 --> node358["node:358"]
362   node179 --> node359["node:359"]
363   node179 --> node360["node:360"]
364   node180 --> node361["node:361"]
365   node180 --> node362["node:362"]
366   node181 --> node363["node:363"]
367   node181 --> node364["node:364"]
368   node182 --> node365["node:365"]
369   node182 --> node366["node:366"]
370   node183 --> node367["node:367"]
371   node183 --> node368["node:368"]
372   node184 --> node369["node:369"]
373   node184 --> node370["node:370"]
374   node185 --> node371["node:371"]
375   node185 --> node372["node:372"]
376   node186 --> node373["node:373"]
377   node186 --> node374["node:374"]
378   node187 --> node375["node:375"]
379   node187 --> node376["node:376"]
380   node188 --> node377["node:377"]
381   node188 --> node378["node:378"]
382   node189 --> node379["node:379"]
383   node189 --> node380["node:380"]
384   node190 --> node381["node:381"]
385   node190 --> node382["node:382"]
386   node191 --> node383["node:383"]
387   node191 --> node384["node:384"]
388   node192 --> node385["node:385"]
389   node192 --> node386["node:386"]
390   node193 --> node387["node:387"]
391   node193 --> node388["node:388"]
392   node194 --> node389["node:389"]
393   node194 --> node390["node:390"]
394   node195 --> node391["node:391"]
395   node195 --> node392["node:392"]
396   node196 --> node393["node:393"]
397   node196 --> node394["node:394"]
398   node197 --> node395["node:395"]
399   node197 --> node396["node:396"]
400   node198 --> node397["node:397"]
401   node198 --> node398["node:398"]
402   node199 --> node399["node:399"]
403   node199 --> node400["node:400"]
404   node200 --> node401["node:401"]
405   node200 --> node402["node:402"]
406   node201 --> node403["node:403"]
407   node201 --> node404["node:404"]
408   node202 --> node405["node:405"]
409   node202 --> node406["node:406"]
410   node203 --> node407["node:407"]
411   node203 --> node408["node:408"]
412   node204 --> node409["node:409"]
413   node204 --> node410["node:410"]
414   node205 --> node411["node:411"]
415   node205 --> node412["node:412"]
416   node206 --> node413["node:413"]
417   node206 --> node414["node:414"]
418   node207 --> node415["node:415"]
419   node207 --> node416["node:416"]
420   node208 --> node417["node:417"]
421   node208 --> node418["node:418"]
422   node209 --> node419["node:419"]
423   node209 --> node420["node:420"]
424   node210 --> node421["node:421"]
425   node210 --> node422["node:422"]
426   node211 --> node423["node:423"]
427   node211 --> node424["node:424"]
428   node212 --> node425["node:425"]
429   node212 --> node426["node:426"]
430   node213 --> node427["node:427"]
431   node213 --> node428["node:428"]
432   node214 --> node429["node:429"]
433   node214 --> node430["node:430"]
434   node215 --> node431["node:431"]
435   node215 --> node432["node:432"]
436   node216 --> node433["node:433"]
437   node216 --> node434["node:434"]
438   node217 --> node435["node:435"]
439   node217 --> node436["node:436"]
```

You can wrap text areas in blocks. The background colours can be defined using colour themes

Author information, like name and institution, gets formatted based on the selected theme. Colours are based on the selected colour theme

You can add title graphics to your title slide. You can set the width and height based on the percentage of the text blocks



Create professional presentations with LaTeX

Customise LaTeX's beamer class to get the professional look you need for your next presentation

Resources

Wikibooks site:

<http://en.wikibooks.org/wiki/LaTeX/Presentations>

Hartwork site:

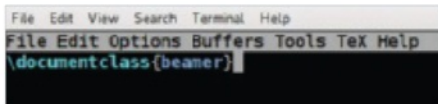
www.hartwork.org/beamer-theme-matrix

LaTeX is a popular text layout program that is used extensively in scientific and technical spheres. Most people have used it when writing theses or articles. What you may not know is that there are document classes that can be used to create presentation slides, too. An older class is called prosper. It has fallen out of use, and has been replaced in many peoples' minds with beamer. Beamer provides all of the structures needed to produce presentations with a relatively

complex layout. There are many tutorials available online to get you started with writing your own presentations. We will cover the basics here in the first half of this article. One common complaint is that the themes available are a bit bland. The second half will cover some of the customisations you can make to the layout and themes to get a more personalised style in your slides. This way, you can get your presentation dressed up to your heart's content.

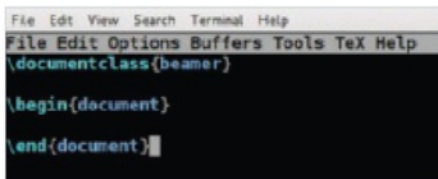
01 Getting beamer

The first step is to get beamer installed on your system. Most distributions will have it included in the package management system. For example, in Ubuntu, you would install it with 'sudo apt-get install latex-beamer'. Your other option is to download the source code from the Bitbucket site and build it yourself.



02 Setting the class

In order to use beamer, you need to define your document class in the preamble. You can set the font size for the text in the presentation. If you want handouts, you can set this option for the document class. While you are writing, you can use the 'draft' option to speed up compiling the slides.

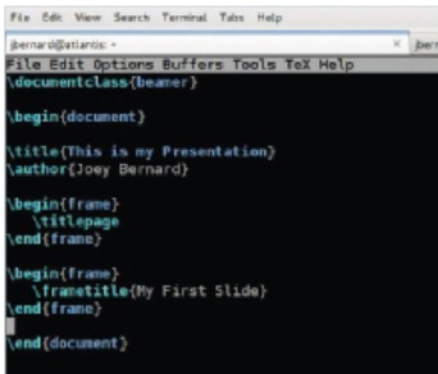


03 Opening a document

Your presentation slides are considered a document by LaTeX. All of the slides will be contained within a section called 'document'. This means that you will need a '\begin{document}' statement and an '\end{document}'. All of your actual slide contents will be between these two statements.

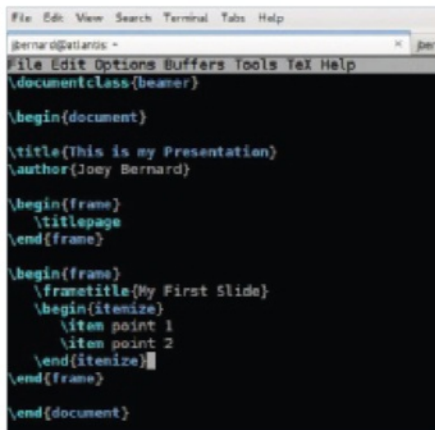
04 Creating a title slide

To generate your title slide, you will need to set the values of various parameters. You can set the '\title{}', the '\author{}' and '\date{}', among other options. Once all of these options are set, you can create a title slide with the command '\titlepage'.



05 Creating your first slide

Slides are defined as sections named frames. They are bracketed by '\begin{frame}' and '\end{frame}' statements. You can set a title for individual slides with '\frametitle{}'. Any options to the frame are added at the end of the '\begin{frame}' line.

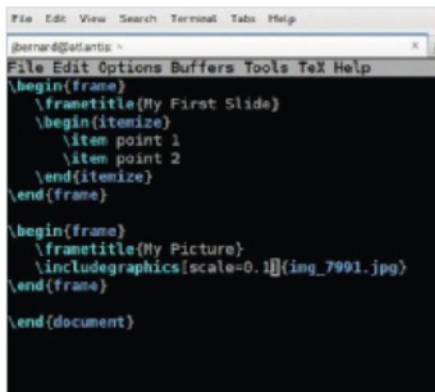


06 Itemised lists

You can add bullet points to your slide by creating an itemised list. This is a section called 'itemize'. Individual bullet points are started with an '\item' command, where the text of the bullet point follows. Beamer handles the proper formatting of text on your slide.

07 Adding images

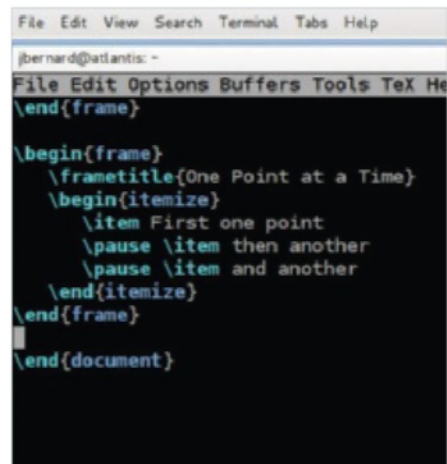
There are several image-handling packages that you can use with beamer. The simplest way to add an image to a slide is with the '\includegraphics' command. You will need to give the path to the image file that you are inserting in this slide.



08 Setting image size

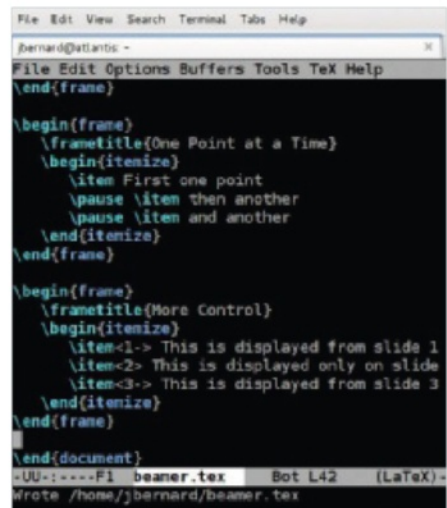
By default, LaTeX simply displays the image full-sized. In most cases, this means

that the image displayed will be larger than the area of the slide. You can add an option to '\includegraphics' to either set the width and height, or scale the image by some factor.



09 Pausing output

When doing a presentation, it is often useful to only present one bullet point at a time. This forces the audience to focus on only the point being discussed. This can be accomplished with the command '\pause'. This causes output display to be paused until a key is hit, usually the space bar.



10 Text animations

Even more control can be had by adding '<a-b>' as an option to the '\item' command. This will tell LaTeX to only display the item in question from slide a until slide b. For example, if you wanted a particular item to be displayed from slide 2 and onwards, you would use '\item<2->'.

11 Adding tables

Tables are a bit messier to include in a slide. You need to open with `\begin{table}` and `\begin{tabular}...`. On the 'tabular' line you need to include cell justification options. You also need to define horizontal lines with `\hline`, and column breaks with `'&'`.

```
File Edit View Search Terminal Tabs Help
bernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\end{frame}

\begin{frame}
\frametitle{More Control}
\begin{itemize}
\item<1-> This is displayed from slide 1 on
\item<2-> This is displayed only on slide 2
\item<3-> This is displayed from slide 3 on
\end{itemize}
\end{frame}

\begin{frame}
\frametitle{My Source Code}
\begin{lstlisting}
int main() {
printf("This is my code\n");
}
\end{lstlisting}
\end{frame}

\end{document}
-UU-:----F1 beamer.tex Bot L49 (LaTeX)
Wrote /home/jbernard/beamer.tex
```

12 Code samples

When doing presentations about programming, you will need to be able to show code snippets on some of your slides. This can be done by using a `lstlisting` section. This section behaves much like a `verbatim` section, in that what you type is what gets displayed, including whitespace. You will also need to add `\usepackage{listings}` to the preamble of your document.

```
File Edit View Search Terminal Tabs Help
bernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\end{frame}

\begin{frame}
\frametitle{More Control}
\begin{itemize}
\item<1-> This is displayed from slide 1 on
\item<2-> This is displayed only on slide 2
\item<3-> This is displayed from slide 3 on
\end{itemize}
\end{frame}

\begin{frame}[fragile]
\frametitle{My Source Code}
\begin{lstlisting}
int main() {
printf("This is my code\n");
}
\end{lstlisting}
\end{frame}

\end{document}
```

13 Fragile frames

If you try to compile the above example, you will get failures. Whenever you use any verbatim-like section, you need to flag that slide as being fragile by adding `[fragile]` to the `\begin{frame}`'s line.

```
File Edit View Search Terminal Tabs Help
bernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\documentclass{beamer}

\usepackage{listings}
\usetheme{Berkeley}
\usecolortheme{albatross}

\begin{document}

\title{This is my Presentation}
\author{Joey Bernard}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\frametitle{My First Slide}
\begin{itemize}
\item point 1
\item point 2
\end{itemize}
\end{frame}

-UU-:----F1 beamer.tex Top L5
Beginning of buffer
```

14 Picking a theme

The default format for a beamer presentation is rather bland. You can add some pizzazz simply by using a canned theme, a canned colour scheme, or both. This can be done by using `\usetheme{}` and `\usecolortheme{}`. Check the beamer theme matrix in the link to see how they interact.

15 Making a PDF

In order to use all of this work, you will need to compile your LaTeX source file into some other file format. If you use the `latex` command,

you will end up with a DVI file. You will then need to convert this file to some other format, such as PostScript or PDF. However, there is a command that can go straight from the LaTeX source file to a PDF file. This command is `pdflatex`. It also takes care of including the graphics package used by `\includegraphics` commands.

```
File Edit View Search Terminal Tabs Help
bernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\documentclass{beamer}

\usepackage{listings}
\usetheme{Berkeley}
\usecolortheme{albatross}

\usepackage{textpos}
\addtobeamertemplate{frametitle}{}{\
\begin{textblock*}[100mm]{\textwidth,-1cm}
\includegraphics[height=1cm,width=1cm,keepaspectratio]{
\end{textblock*}}
\end{document}

\begin{document}

\title{This is my Presentation}
\author{Joey Bernard}

\begin{frame}
\titlepage
\end{frame}

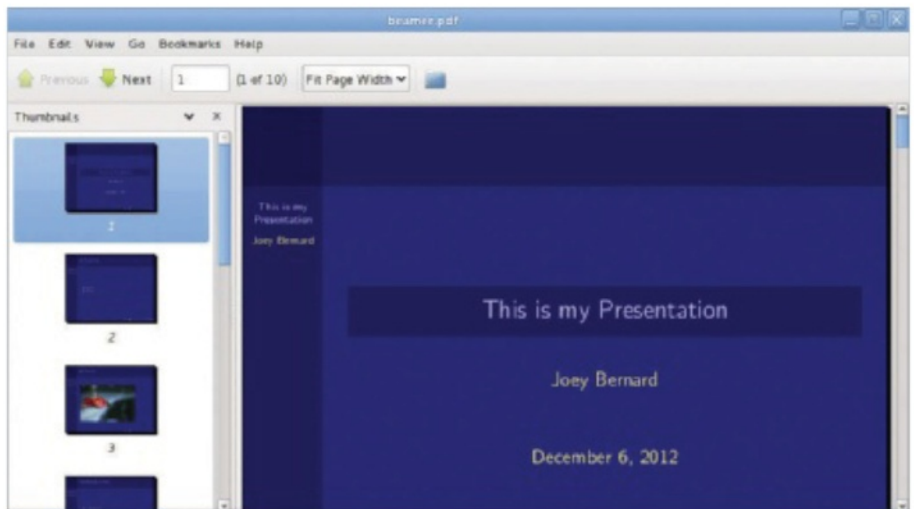
-UU-:----F1 beamer.tex Top L7 (LaTeX)
Wrote /home/jbernard/beamer.tex
```

16 Adding a logo

In order to add a logo, you can override the definition of a frame title with `\addtobeamertemplate`. This command needs to go in the preamble and can include text labels, as well as an image file for a logo.

17 Adding a background

You may want to include an image as a background to your slides. This can be done with the command `\setbeamertemplate{background canvas}`, where you can set the background with commands like `\includegraphics`. You can also set other template options, like various colours.



```

File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\documentclass{beamer}

\usepackage{listings}
\usepackage{Berkeley}
\usecolortheme{albatross}
\usecolortheme[RGB={205,173,0}]{structure}

\usepackage{textpos}
\addtoamertemplate{frametitle}{\%
\begin{textblock*}{100mm}{\textwidth,-1cm}
\includegraphics[height=1cm,width=1cm,keepaspectratio]{img_7991.jpg}
\end{textblock*}}

\setbeamertheme{background canvas}{\includegraphics
[width=\paperwidth,height=\paperheight]{img_7991.jpg}}

\begin{document}

\title{This is my Presentation}
\author{Joey Bernard}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\frametitle{My First Slide}
\begin{itemize}

```

18 Changing the default colour
 Instead of using one of the preset colour themes, you can specify the colours to use directly with RGB values. As part of the ‘\usecolortheme’ command, you can set the RGB values and tell LaTeX to assign it to the structure portion of your slides.

19 Changing itemisation markers
 By default, beamer uses triangles to mark each bullet point on your slide. You can change these to balls, circles or rectangles. You need to include the command ‘\setbeamertheme{items}[option]’, where you replace ‘option’ with the marker you wish to use.

“Instead of using a preset colour theme, you can specify the colours to use directly with RGB values”

```

File Edit Options Buffers Tools TeX Help
\end{textblock*}}

\setbeamertheme{background canvas}{\includegraphics
[width=\paperwidth,height=\paperheight]{img_7991.jpg}}

\setbeamertheme{items}[ball]
\setbeamertheme{blocks}[rounded]

\begin{document}

\title{This is my Presentation}
\author{Joey Bernard}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\frametitle{My First Slide}
\begin{itemize}
\item point 1

```

20 Adding rounded corners
 When you include blocks on your slides, you can tell beamer whether or not to use rounded corners. Again, you need to use the ‘\setbeamertheme’ command with the ‘blocks’ option. You can hand in the sub-option ‘rounded’ in square brackets. This will then apply to all block structures in the file.

21 Drop shadows
 Drop shadows are a common graphical technique to add more interest to a display. This is an extra option to the ‘\setbeamertheme’ command, applied to the ‘blocks’ sections. This option can be combined with the rounded corners option from above. Or, it can be used instead of the rounded corners option.

```

File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\end{textblock*}}

\setbeamertheme{background canvas}{\includegraphics
[width=\paperwidth,height=\paperheight]{img_7991.jpg}}

\setbeamertheme{items}[ball]
\setbeamertheme{blocks}[rounded][shadow=true]
\setbeamertheme{navigation symbols}{}

\begin{document}

\title{This is my Presentation}
\author{Joey Bernard}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\frametitle{My First Slide}
\begin{itemize}

```

22 Removing navigation symbols
 Beamer includes a set of navigation symbols at the bottom of the slides, allowing you to move through your slide deck. Depending on style issues, you may want these removed from the end product. To do so, you can use the ‘\setbeamertheme’ command against the ‘navigation symbols’ section. You simply need to hand in an empty list to remove the symbols.

```

File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~
File Edit Options Buffers Tools TeX Help
\end{textblock*}}

\setbeamertheme{background canvas}{\includegraphics
[width=\paperwidth,height=\paperheight]{img_7991.jpg}}

\setbeamertheme{items}[ball]
\setbeamertheme{blocks}[rounded][shadow=true]
\setbeamertheme{navigation symbols}{}
\useoutertheme{infolines}

\begin{document}

\title{This is my Presentation}
\author{Joey Bernard}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\frametitle{My First Slide}
\begin{itemize}

```

23 Adding a custom footer
 Some themes provide a footer line that contains information like the slide number, the name of the author, the title of the presentation and other details. However, not all themes do. You can add it to themes that don’t normally have it by using the command ‘\useoutertheme{infolines}’.

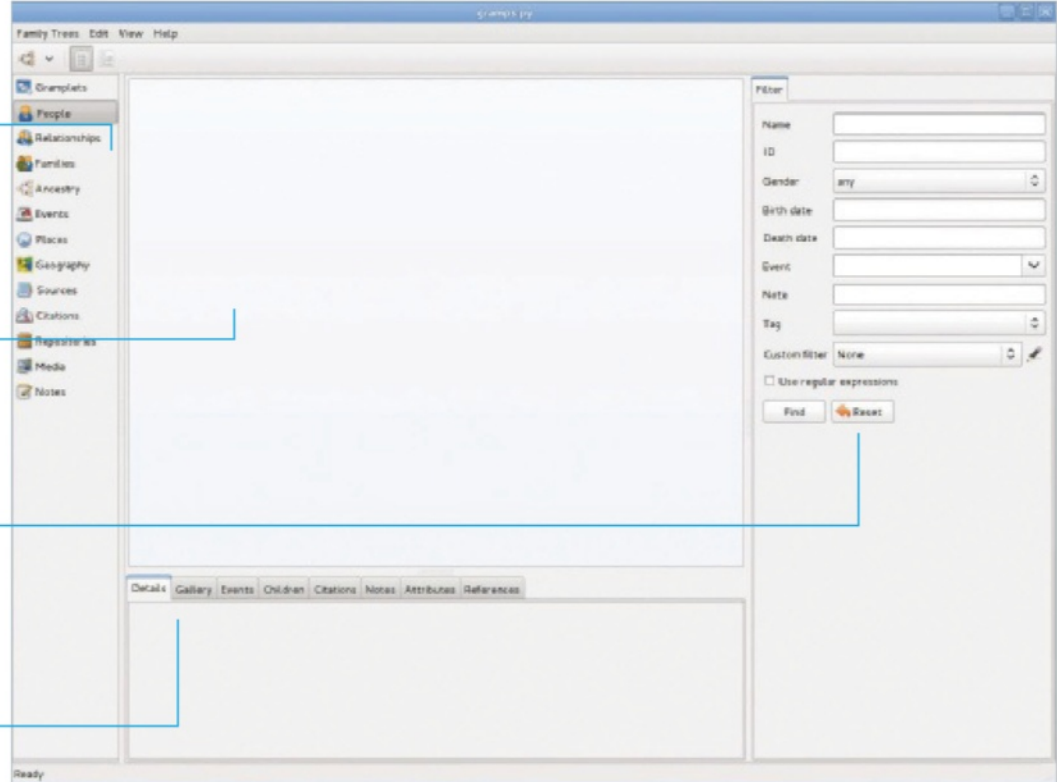
24 Where to now?
 As you can see, almost all options in beamer are customisable. We’ve only had time to look at a few of the most common options that most people will be interested in. But looking at the documentation, you will be able to make beamer do exactly what’s needed to make the presentation look and feel the way you want.

This panel provides quick access to all the various sections, like people, events, places and media entries

The centre panel will show lists of whatever object type you have selected in the left side panel, like a list of people if you select the 'People' entry

If you have many entries, you can filter down to a manageable subset by selecting filter criteria here

You can find details for the item you select above. The tabs that are available depend on what type of object you are looking at



Make the most of open-source genealogy with Gramps

Genealogy is a fun hobby for many people around the globe. While there are lots of programs for Windows, Linux options have been lacking. Gramps helps fill this gap

Resources

Gramps: gramps-project.org

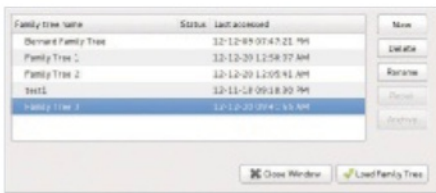
Genealogy is a very popular hobby with people all around the world. If you pick up any of the genealogy magazines at your local bookstore, they will cover Windows programs that are available. But no one seems to have realised that people who use Linux are also interested in researching their family trees. While there have been text-based programs to work with GEDCOM files, there hasn't been a really complete GUI program until Gramps came onto the scene. With Gramps, you have all of the functionality available

that Windows users have enjoyed for some time. Gramps is written in Python and is designed with a plug-in architecture. This means that you can rather easily write your own plug-in to add any specialised functionality that isn't provided through a base installation. In this article, you'll see how to get started on your family tree, analyse the data that you have collected, and generate reports that you can print and share with others. You'll also learn how to share the data you have collected through GEDCOM files.



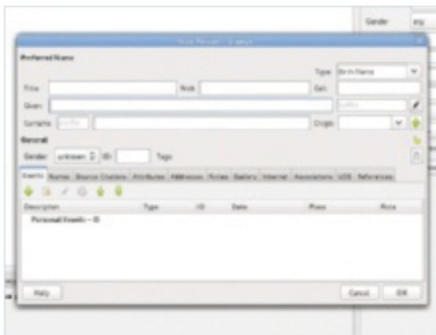
01 Getting Gramps

Gramps should be available in most Linux distributions. For Debian-based ones, you would install with 'sudo apt-get install gramps'. There are Windows binaries available, if you want to get your friends hooked on Gramps, too. As always, the source is available if you would rather compile your own copy.



02 Create a new tree

The first step, once you have Gramps, is to create a new family tree. This can be considered almost like a project, for the software developers out there. A family tree will contain all of the information relevant to a single lineage going back in time.



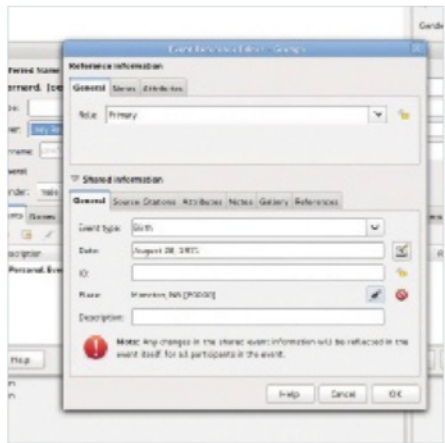
03 Add a new person

A family tree is of little use without people. So, the next step is to add your first person. If you are doing your own family tree, you may as well start with yourself. Clicking

on the plus sign in the icon bar will bring up a dialog box where you can enter all of the relevant information.

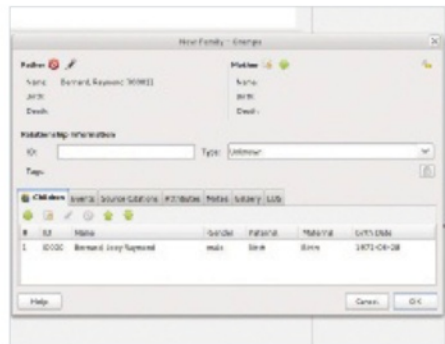
04 Setting the home person

To generate charts and reports from all of this data, you will need to set a 'home person' who will act as the root of the family tree. Again, if you are doing your own family tree, it would make sense to select yourself. To do so, you will need to select the People entry on left-hand side of the screen. Then select the person from the list, and click on Edit>Set Home Person from the main menu.



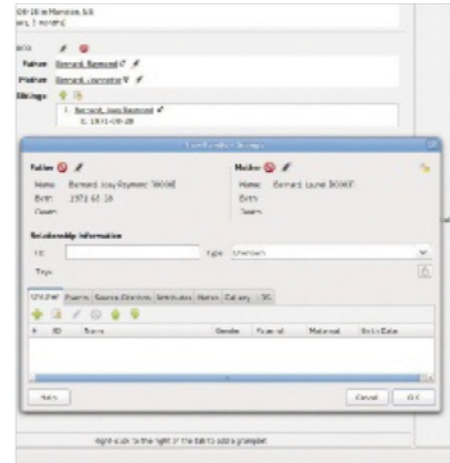
05 Adding birth information

Double-clicking a person from the list will reopen the Edit dialog. At the bottom, you can add events to the person's record. Arguably the most important event is their birth. To add an event, click on the plus icon in the event section.



06 Adding parents

Click on the Relationships entry on the left-hand side. This will open the relationship section for the selected person. You can add parents by clicking on the plus icon in the top bar. You can either add existing people or create new people, for the father and mother.

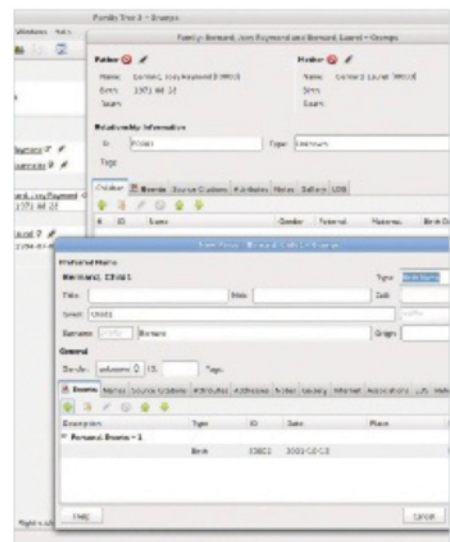


07 Adding a spouse

Staying in the Relationships section, you can click on the 'Add a new family' icon to create a new family with this person as a parent. You can then either add an existing person or create a new person as the spouse.

08 Setting marriage information

While the New Family dialog is open, you will need to click on the Event tab in the bottom section. Click on the plus icon to add a new event, selecting Marriage as the type. You can then set the date and location.

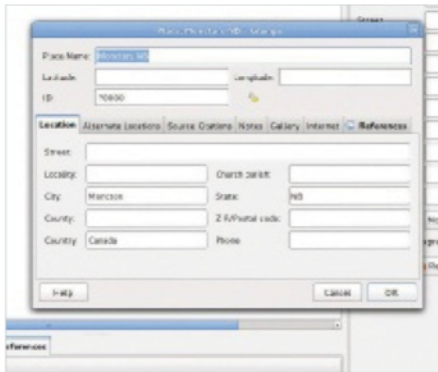


09 Adding children

Before closing the New Family dialog, you can also add any children. Clicking on the Children tab in the bottom section will allow you to add either existing people or new ones as children of this family unit.

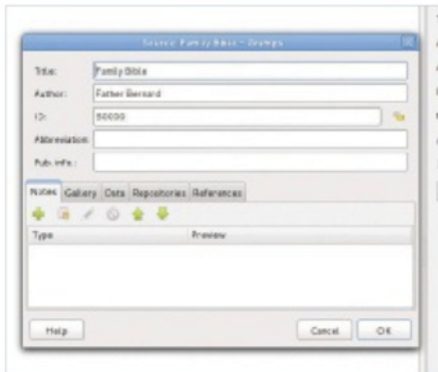
10 Editing events

You have used events to add information, like births and marriages. There are many built-in event types when you install Gramps, covering most areas of life. You can edit the details of any of these events by clicking on the Events link on the left-hand side.



11 Adding and editing places

Events take place somewhere in the real world. When you add a new event, you can select a place where it happened. After you have added a place to an event, you can edit the details by clicking on the Places link on the left-hand side and selecting the place of interest.

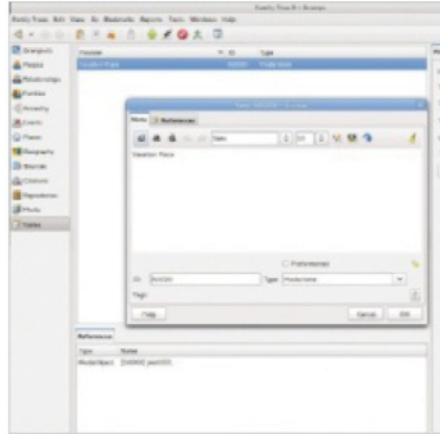


12 Adding sources

You can add sources to your family tree by clicking on the Sources link on the left-hand side. You can add a new one by clicking the plus icon and filling in all of the relevant information. Recording where you get your information is a very important part of research.

13 Adding media

In this age of digital media, it is easier than ever to add a new dimension to your research. You can add media objects to your family tree. Once added, they can be linked through references to events and people.



14 Adding notes

All of the objects in Gramps can have notes attached to them. This allows you to add extra information to items like photos that have been added, places of importance, sources of information. You can edit details of these notes by clicking on the Notes link.

15 Displaying your ancestral chart

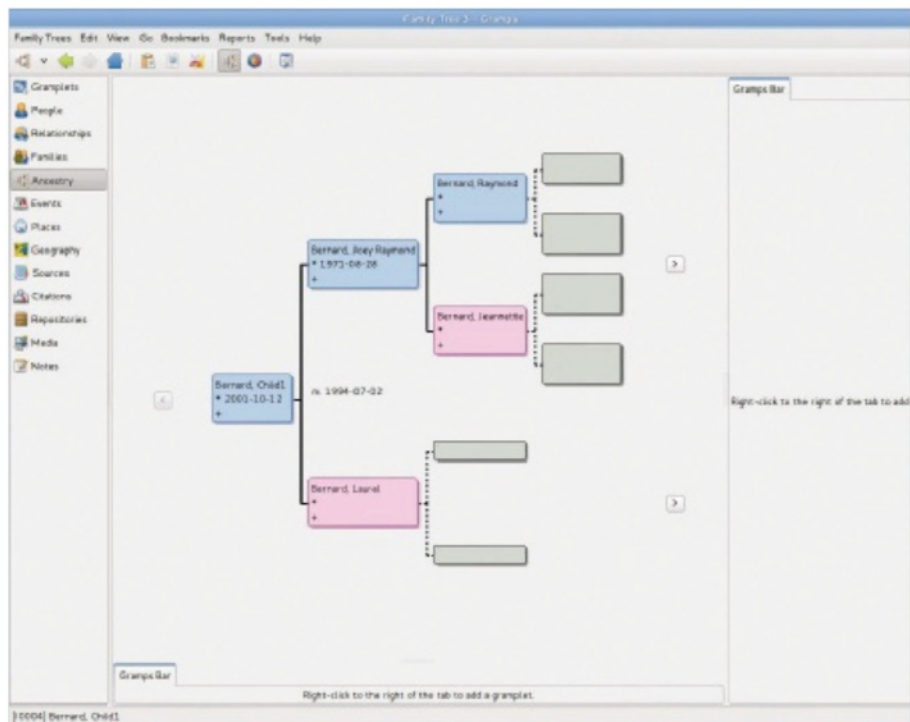
When you have all of your information entered into Gramps, you can start to see what your family tree looks like. There are many charting functions available. They are available under the Reports menu item. The usual chart

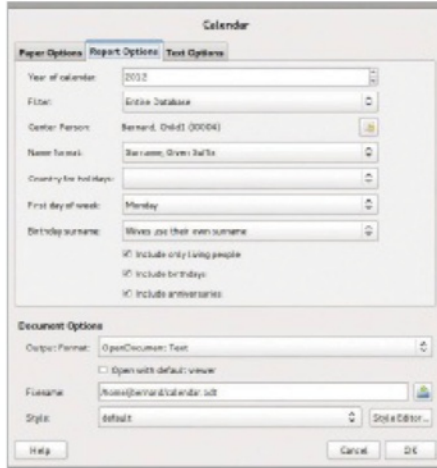
“When you have all of your information entered into Gramps, you can start to see what your family tree looks like”

that you will likely want to look at is the ancestral chart. This shows all of your direct ancestors in the classic family tree form. Clicking on the menu item Reports>Graphical Reports>Ancestor Tree will open a dialog window. Here, you can select how many generations to display and presentation details.

16 Displaying a fan chart

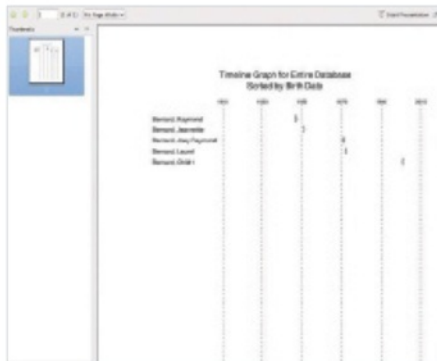
Another way of visualising your ancestors is with a fan chart. You can select the number of generations to look at, and several display options. These graphs are interactive, allowing you to move around in your history.





17 Generating a calendar

With the power of Python, you have the ability to do very complicated processing. One of the included plug-ins allows you to generate a calendar of important dates, like birthdays and marriage anniversaries. You can generate a calendar by clicking on Reports>Graphical Reports>Calendar.

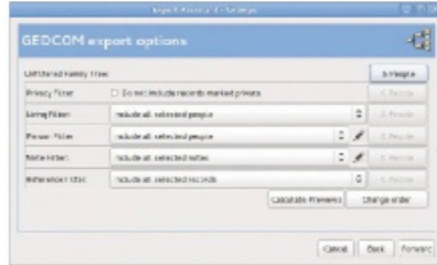


18 Looking at a timeline

You can generate a timeline of events throughout your history. Clicking on Reports>Graphical Reports>Timeline Chart will open a new dialog where you can set the options. By default, it will create an OpenDocument file, but this can be changed to one of several other file formats.

19 Generating a website

You can generate a full website, with navigation, of all of your genealogical information. You can choose how much information gets exported, and details of the formatting. You can set the stylesheet that is used for the display. You can even export the whole thing into a zipped tarball.

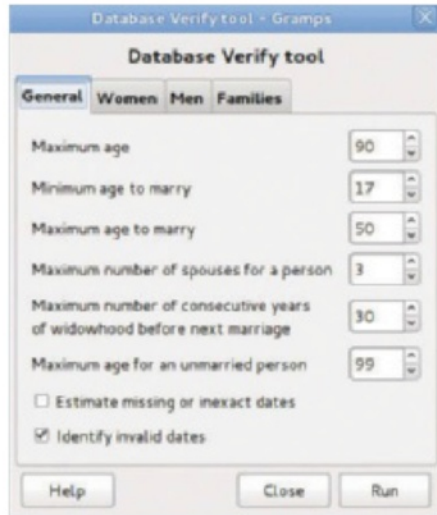


20 Exporting a GEDCOM file

The standard file format for genealogical information is called GEDCOM. There are several versions around, the latest one being 5.5. Gramps exports to the latest version. This is how you can share the information you've collected with other researchers.

21 Checking an import

You can also import data from a GEDCOM file. Sometimes these files may have errors in them. You can check the imported data by clicking on the menu item Tools>Family Tree Repair>Check and Repair Database. This will check several possible problems in your data.

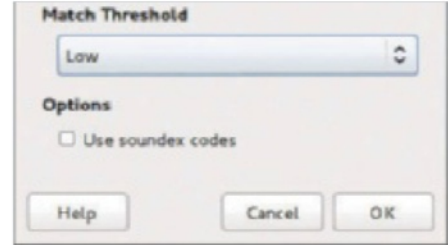


22 Verifying your data

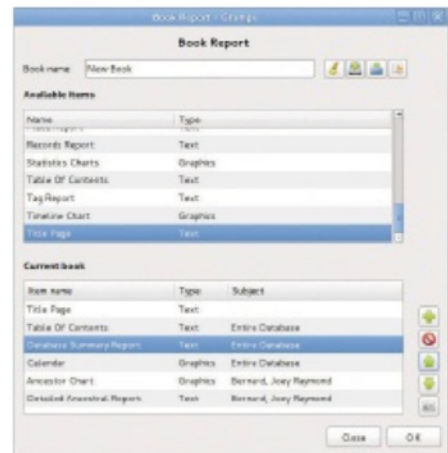
You may also need to verify the data itself. Do you have a 200-year-old relative? Do you have someone who has kids that are older than they are? Clicking on Tools>Utilities>Verify the Data will bring up a dialog window where you can set the parameters to use in the verification.

23 Locating duplicates

One common issue, especially when importing data from other sources, is having duplicate entries in your database. Clicking on

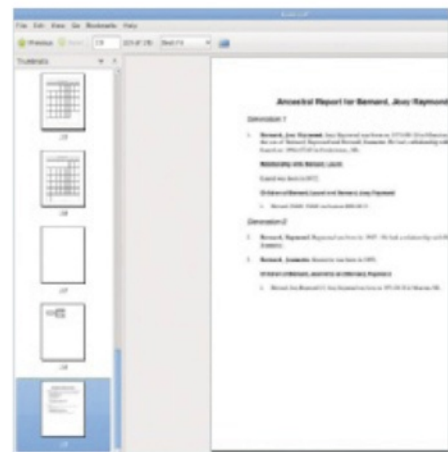


Tools>Family Tree Processing>Find Possible Duplicate People will bring up a dialog box where you can set the sensitivity. Then it will search through the data for possible matches.



24 Getting a full book

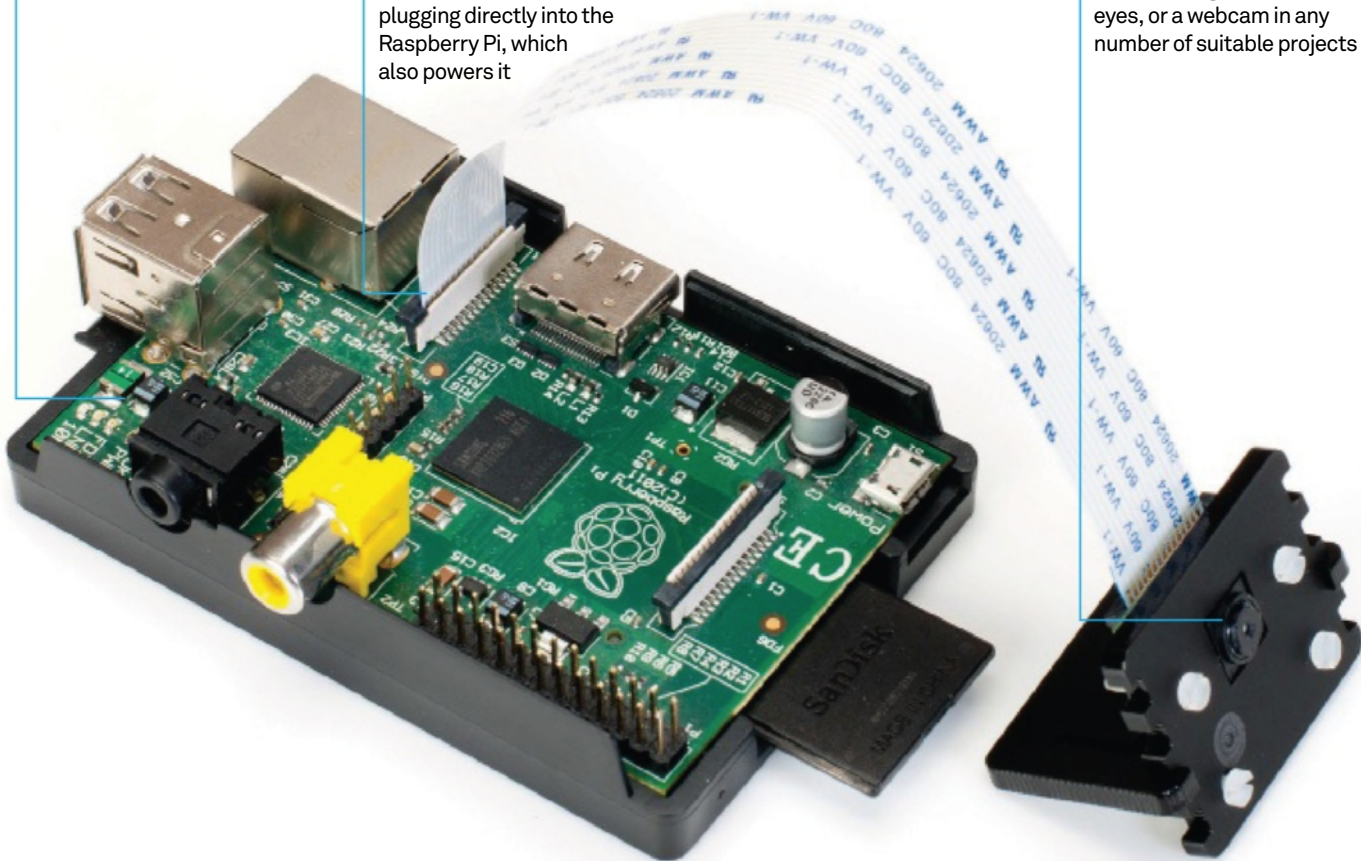
You can generate a full book report containing all of the information you have collected. Clicking on Reports>Books>Book Report will open up a window where you can select which items to include. Each of these elements can be configured individually. Once everything is ready, you can click on 'OK' to get output in either an ODT file or a PDF file.



Create automated video- and photo-related tasks easily

The Pi Camera doesn't take up any USB slots, instead plugging directly into the Raspberry Pi, which also powers it

Use the camera for time-lapse photography, a robot's eyes, or a webcam in any number of suitable projects



Take pictures and video with the Raspberry Pi Camera

Follow our short tutorial on how to get a Pi Camera set up on your Raspberry Pi, and how to use it

Resources

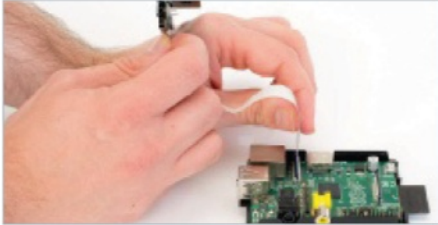
Raspberry Pi Raspbian:
www.raspberrypi.org/downloads
Pi Camera
Ashton's picam module:
<https://github.com/ashtons/picam>

One of the most recent Raspberry Pi accessories is the tiny Pi Camera board – a small PCB with a camera sensor mounted to it that connects via a ribbon to the Raspberry Pi. Because of this, it's not exactly plug-and-play, so you'll need to do some extra setup on your Raspberry Pi to get it to work.

The Pi Camera has multiple functions, such as for time-lapse photography, using as a webcam, or even as an optical sensor for a Pi-

powered robot, like the one we're making this issue. Because it doesn't take up any USB slots, and draws very low power, it can be a lot more versatile than a standard webcam.

The Pi Camera itself is not a low-quality piece of kit either – with a 5MP sensor, it's also able to create up to 1080p quality video, the same as the Raspberry Pi's HDMI output. So grab your Raspbian SD card and get started making the most out of your Pi Camera.



01 Attach Camera To attach the Camera to the Raspberry Pi, locate the slot between the Ethernet and HDMI port and lift up the fastener. Insert the ribbon of the Camera board, making sure to align the ribbon's connectors with those on the Pi.



02 Pi preparation Before we try to enable the Raspberry Pi Camera, we need to make sure our firmware and software are all up to date with a quick software upgrade. In Raspbian, we do this by opening the terminal and using:

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

...followed by:

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

03 Pi config Once that's finished, run in the terminal or command line:

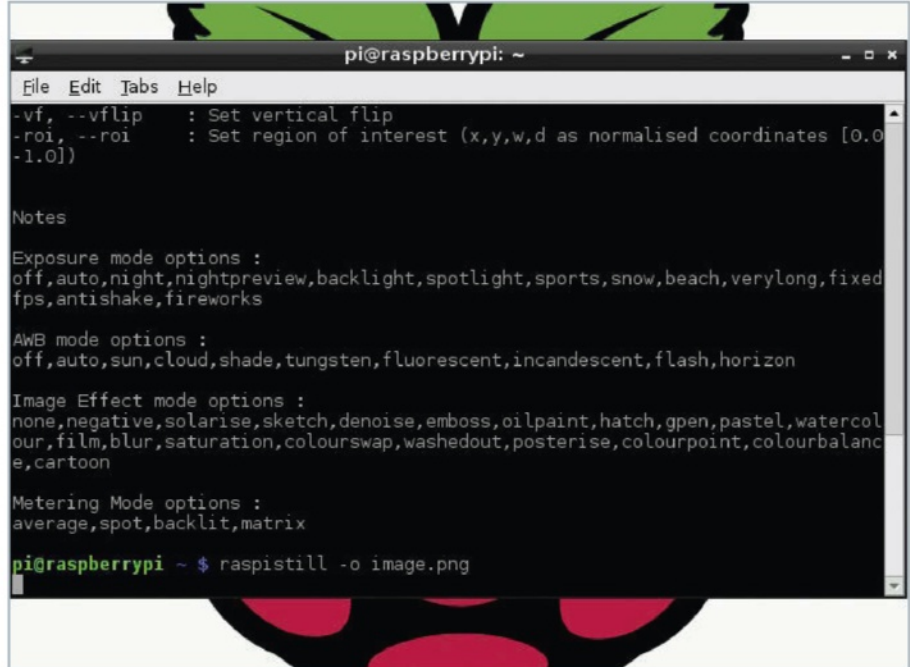
```
$ sudo raspi-config
```

...to start the standard configuration screen. Navigate down to Enable Camera, press Enter and then simply key over to enable and confirm with another press of Enter. Select Finish and then reboot.

04 Take pictures To take pictures with the Raspberry Pi Camera, you'll simply need to enter:

```
$ raspistill -o image.png
```

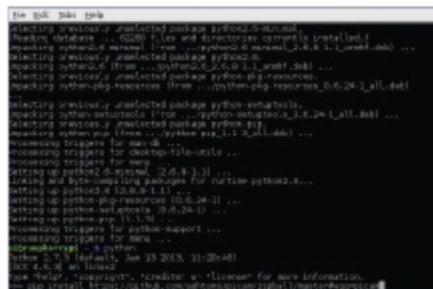
This will show a five-second preview of the input of the camera and then capture the last frame of the video.



05 Record video To record a video, we use a similar command, raspivid, like so:

```
$ raspivid -o video.h264
```

It will also take five seconds of video by default.



06 Picam If you want to do a little more with the Pi Camera, there's a simple Python wrapper currently available called picam. You'll need to install it first, though, and we'll use pip for that. Install pip with:

```
$ sudo apt-get install python-pip
```

...and then enter:

```
pip install https://github.com/ashtons/picam/zipball/master#egg=picam
```

07 Picam photos We can now use Python to construct a script to take photos with the picam module. Very simply, all you need to do is enter:

```
import picam
i = picam.takePhoto()
i.save('/home/pi/test.jpg')
```

And running it will take a photo called test.jpg.

08 Advanced photos You can have it take photos of specific size and quality with a time-based name by editing the code to look like this:

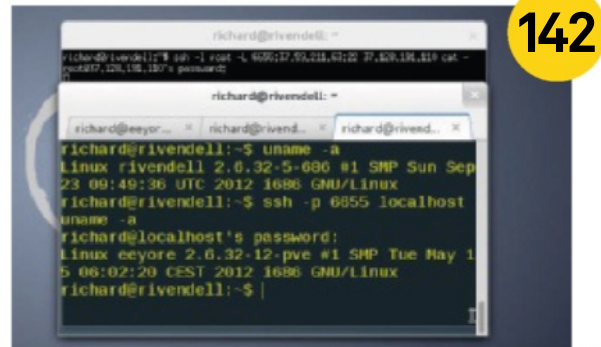
```
import picam
import time
ii = picam.takePhotoWithDetails(640, 480, 85)
filename = "/tmp/picam-%s.jpg" % time.
strftime("%Y%m%d-%H%M%S")
ii.save(filename)
```

09 Picam video and more Picam also allows you to take video in a similar way to the above, with the main difference being that you'll use the recordVideo command. You can use the code to take photos or video at regular intervals for time-lapse, or have it trigger during a specified event.

Hacks

Customise and tweak your system

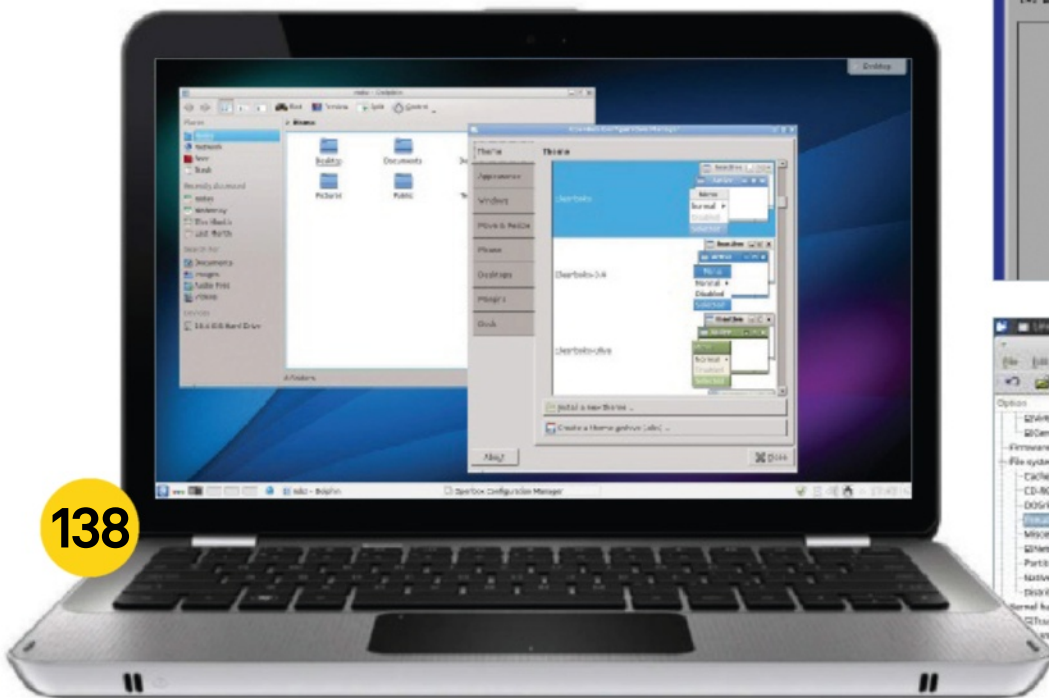
- 132** **Turbocharge your cloud**
Use the power and efficiency of a lightweight solution
- 138** **Speed up Linux with Openbox**
Speed up your day-to-day computing without sacrificing usability
- 142** **Bypass restrictive firewalls using SSH tunnelling**
Create secure network connections on the fly
- 146** **Create a custom build of Gentoo**
Build a custom distro from the ground up



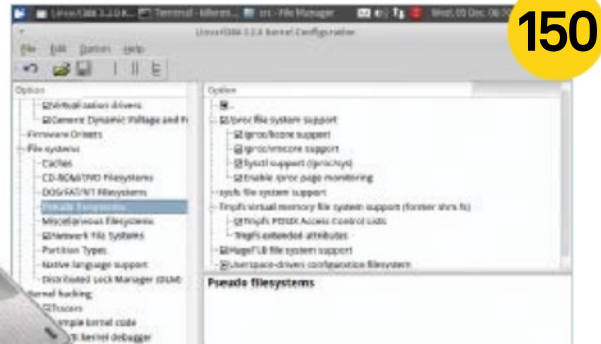
142



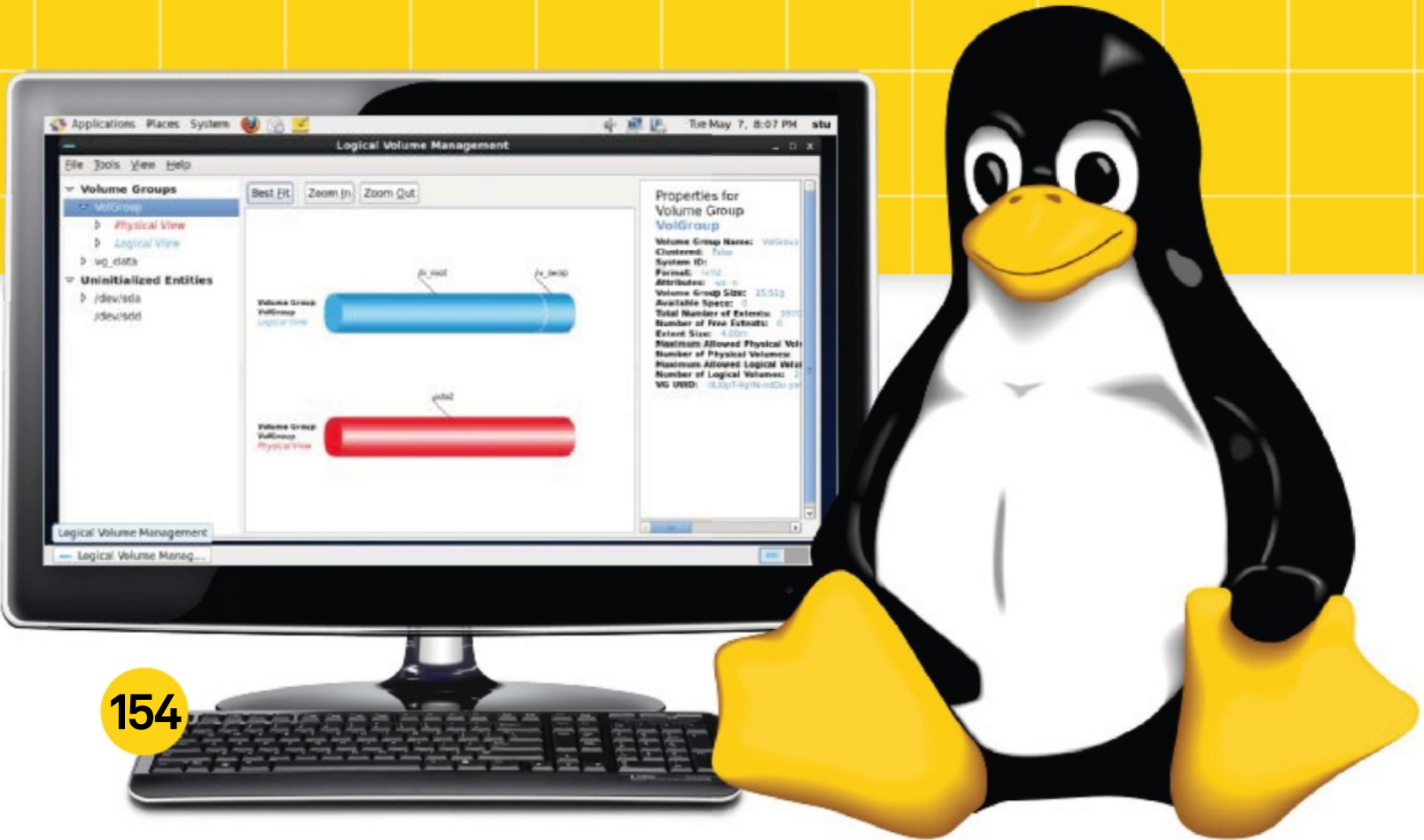
146



138



150



154

“We’ll take you through the steps you need to compile your own customised kernel”

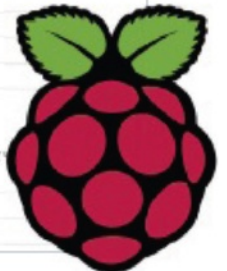
150 Create a custom Linux kernel to optimise performance
Compile your own customised kernel for performance and specialised use

154 Resize your disks on the fly with LVM
Never reformat and restore your drive again

158 Scrape Wikipedia with BeautifulSoup
Parse Wikipedia’s HTML and store it for offline reading

162 Turn your Raspberry Pi into a secure router
Surf the internet privately with Onion Pi

162



Turbocharge your CLOUD

Build your own cloud services using the power and efficiency of a lightweight solution

The value of an efficient web server comes into its own when serving cloud applications over the web or an internal network. This is because every increase in efficiency increases the total number of clients to which you can simultaneously provide services. For this reason, this project employs Nginx, a lightweight, performance-optimised web server, rather than the more common Apache. In most cases, Nginx requires extra configuration to support these services, and we'll cover how you go about doing that. In this example, we're going to use a fresh installation of CentOS 6 as the

host operating system. This tutorial makes use of CentOS 6.4, but the instructions should be much the same for Red Hat and Fedora.

Once we have the Nginx server fully working, we shall add some typical cloud applications. WordPress is a blogging application and relatively easy to install. Feng Office Community Edition is an open source business application that offers word processing alongside team-orientated facilities such as planning and time tracking. OwnCloud is a file sharing application at heart, but offers some other nice facilities such as a media browser.

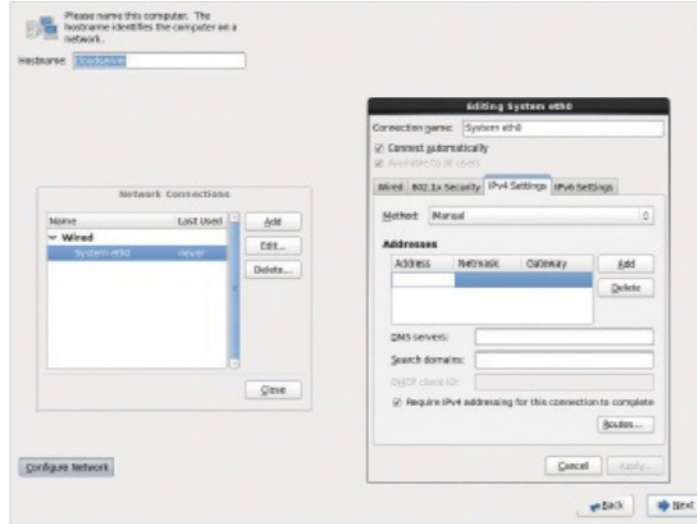
Set up the server OS

Begin by installing CentOS. While installing, you can accept most of the defaults, but it's a good idea to customise the network setup during installation. Firstly, give the computer a meaningful hostname such as 'cloudserver'. On the same page of the installation, select the Configure Network button in order to bring up the Network Connections dialog. Select the network adaptor that you are going to use to connect to clients on your LAN or the internet and then click on 'Edit...'. The first thing to select here is the 'Connect automatically' option. It's typical to set a static address in the case of a server rather than allowing DHCP to assign one automatically. To do this, select the IPv4 tab and change the 'Method:' drop-down from DHCP to Manual. Now click on Add and add an IP address. This should be congruent with the address layout of the rest of your network. So, for example, if your machines have an IP address that begins 192.168.0.x, an IP address of 192.168.0.100 with a netmask of 255.255.255.0 and a gateway 192.168.0.1 should be suitable. Typically, the DNS server address provided by your ISP should be entered. Click Apply when finished.

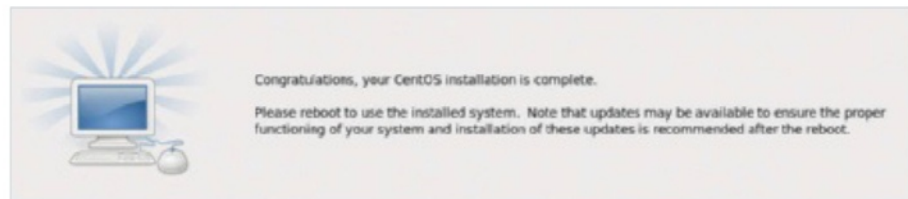
We recommend selecting the 'desktop' installation profile, presuming you want a system with a GUI. After rebooting, log in, open a terminal and type `su` to become root. You will have to carry out nearly all of the operations in this tutorial as root.

Install server components

We'll add the the Nginx repository to the system next. To manually add the source repository, create a file called `/etc/yum.repos.d/nginx.repo`



■ Setting the network options for static IP while installing CentOS



■ CentOS installed and ready for reboot

and add the following:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/6/$basearch/
gpgcheck=0
enabled=1
```

We need to configure the way that Nginx accesses PHP-FPM. Add these lines:

```
location ~ \.php$ {
```

TIP **How to pronounce Nginx:**
If it ever comes up in a real-life conversation, say it like 'engine ex'.

```
root /usr/share/nginx/html;
try_files $uri =404;
fastcgi_pass 127.0.0.1:9000;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
include fastcgi_params; }
```

Make the services permanent

Ensure that the MySQL server (named mysqld), PHP-FPM (named php-fpm) and Nginx (nginx) services start without errors by typing `service [service name] start`. Once this has been checked, type `chkconfig --add [service name]` to cause them to start on boot. Type `service --status-all` to list all running services. This list is quite long, and you can filter it using `grep` like this: `service --status-all | grep sql`. You can find more detailed information about a running service by typing `service [service name] status`.



■ Creating a MySQL database. A handy skill to have when working with a web server

Add webmail

If you are offering services via the cloud, you might already be using webmail such as Gmail. However, if you have tight requirements for how people use email, you could consider adding your own front-end. This usually takes two possible forms. You can add your own IMAP server such as Cyrus (cyrusimap.web.cmu.edu) or Dovecot (www.dovecot.org), both available from the CentOS package repository. You then install a IMAP web front-end into Nginx using the techniques that we've already covered, alongside the specific documentation for those packages. SquirrelMail (squirrelmail.org) is a lightweight front-end and Roundcube (roundcube.net) is more fully featured.

The second approach is to host an IMAP front-end, but then use a webmail service such as Gmail as the IMAP back-end rather than running the server yourself. This allows you to customise the experience along with all the benefits of off-site email.



■ Logging into SquirrelMail

TIP Fetching the latest WordPress

The latest WordPress version is always available from a fixed URL (wordpress.org/latest.tar.gz). We wish more open source projects would do this!

Install cloud applications

WordPress

Let's provide our users with blogging facilities using WordPress. This is good first test for our server because WordPress is extremely well documented and relatively easy to install. If this won't install, something has gone seriously wrong with the setup, and yet it's extensive enough to fully test the server. Go to the WordPress site and download the latest version. If you prefer, you can always directly obtain the latest version of WordPress by typing `wget http://wordpress.org/latest.tar.gz`. Once you have it, type `tar -xvzf [name of archive] -C [output directory]` to unpack it. We can find the root directory of the web server by locating the line 'root' inside default.conf. By default it should be '/usr/share/nginx/html;', and this is the directory in which we need to unpack the archive. Before we can begin the installation, we need to set up the MySQL database that it needs.

Create MySQL database

Experience with MySQL is one of the most useful skills to have when working with web servers. Creating a MySQL database and user is a procedure that we will be repeating during this tutorial. Typically, each cloud application should use its own MySQL database with its own user. We use the Root user account that you set up earlier on to administer all of these users and databases. Note that this 'Root' user has no relationship to the privileged user account that you use to administer Linux systems;

TIP Downloading CentOS

CentOS is now provided as two DVD images. It might be a good idea to give preference to using a torrent, which is usually the fastest source and located next to the ISOs in the repository.

they just happen to have the same name. Restart the MySQL server by typing `service mysqld restart`. We're going to use 'mysql', the command-line tool, to create the MySQL database. The syntax of MySQL queries look quite intimidating, but they're actually fairly easy to work with. Some tutorials create the user of each database with a separate command but it's not really needed for this work, and we will just use the GRANT command. We create a new database like so:

```
CREATE DATABASE [name of database];
```

We set up a new user like so:

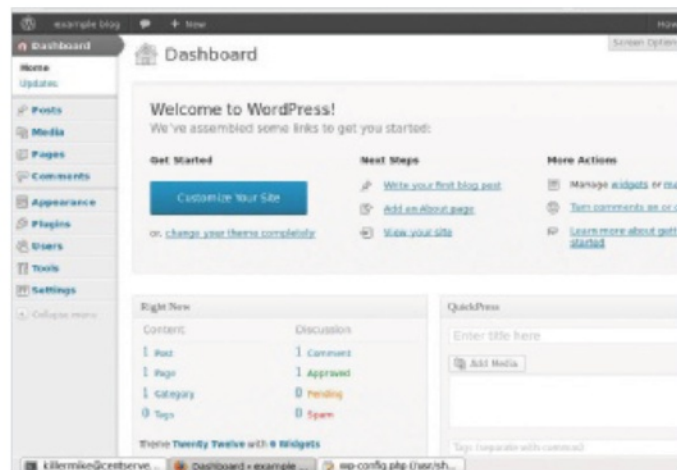
```
GRANT ALL PRIVILEGES ON [database name.*] TO "[name of user]"@"[domain (usually localhost)]" -> IDENTIFIED BY "[password]"
```

As mentioned earlier, this creates a new user as well as assigning a password to it. Now let's get to work setting up a database and user for WordPress.

Log into the MySQL client by typing `mysql -u adminusername -p`. You will be prompted to enter your password and when you have, you should find yourself on the 'mysql>' prompt. Type `CREATE DATABASE wordpress;` to create the database. This should output: `Query OK, 1 row affected (0.01 sec)`



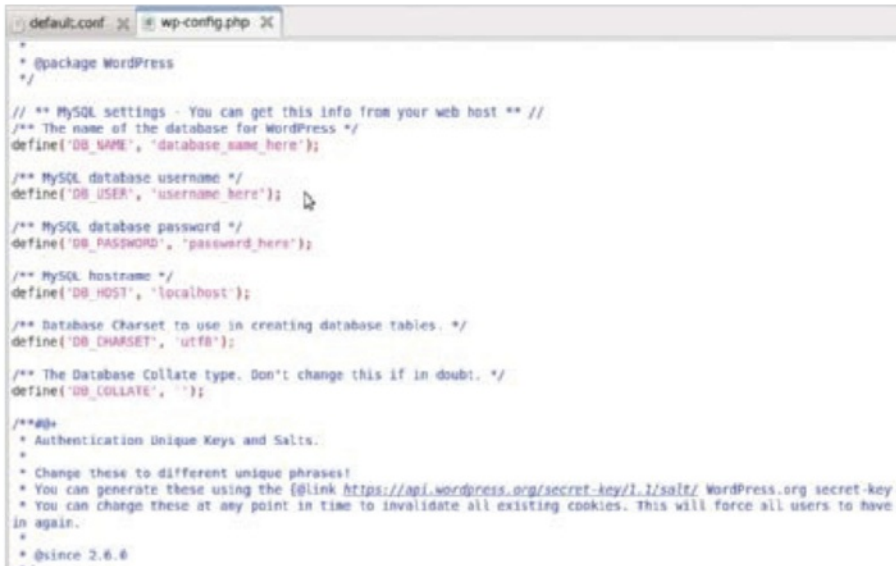
■ Adding the MySQL database settings during WordPress installation



■ Using the WordPress Admin panel, having logged in for the first time



■ The Feng Office word processor is highly integrated with the other tools ■ Entering the database details in the Feng Office installer



■ Manually adding the Nginx repository using nano as the text editor

Type `GRANT ALL PRIVILEGES ON wordpress.* TO "wordpressuser"@"localhost" -> IDENTIFIED BY "password";` substituting the password for one of your own choosing. This creates the user and assigns a password. Note the semicolons that end each command. The response should be:

```
Query OK, 0 rows affected (0.00 sec)
```

Now type in `FLUSH PRIVILEGES;` followed by `quit;`

Complete installation

We'll now edit the WordPress configuration and then run the script that completes the installation. Make sure that you are in the `/usr/share/nginx/html` directory before starting. Create a copy of the example configuration file by typing `cp wordpress/wp-config-sample.php wordpress/wp-config.php`. Now

open the file in a text editor. Set the fields 'DB_NAME', 'DB_USER' and 'DB_PASSWORD' so that they equal, respectively, the name of the MySQL database that you created (wordpress), the name of the WordPress user that you created in the database (wordpressuser) and the password that you created for that user. Save the file.

Now navigate your web browser to `localhost/wordpress/wp-admin/install.php`. If everything has gone okay, you should now receive a message asking you to create a default site. Call it anything you like, as we can change it later.

TIP Make a symbolic link

If you prefer, you can make a symbolic link to the web server directory to save on typing. For example, `ln -s /usr/share/nginx/html /webserve` (note the space) allows you to just type `'webserve'`.

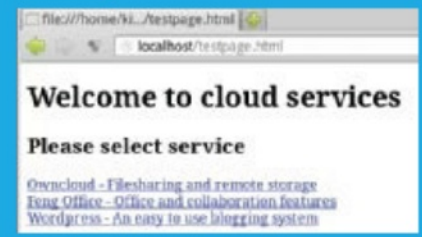
TIP Note all passwords

We'd advise you to create an empty document called 'my passwords' at the beginning of this project, as you're going to have to record quite a lot of usernames, passwords and database names.

Create an entry page

Create an entry page for your services. Here's an example. Replace 'localhost' with the domain name or IP address of your server when you're ready to take it out of testing and make it accessible to other machines. Save this as `index.html` and then place it in `/usr/share/nginx/html/`, the web server root.

```
<!DOCTYPE html>
<html>
<body>
<h1>Welcome to cloud services</h1>
<h2>Please select service</h2>
<a href="http://localhost/owncloud/">Owncloud - Filesharing and remote storage</a><br>
<a href="http://localhost/feng/">Feng Office - Office and collaboration features</a><br>
<a href="http://localhost/wordpress/">Wordpress - An easy to use blogging system</a><br>
</body>
</html>
```



Enter an admin password and your email address and then click on Install WordPress. Log into the site with your username and password as a final check that everything is working okay.

Feng.net Community Edition

Feng Office offers an impressive suite of office facilities, including word processing, time management, contact management, calendar and presentation creation among many other features. It's all tied together in an integrated suite that encourages team collaboration in a business context. We'll be working with the free, open source Community Edition.

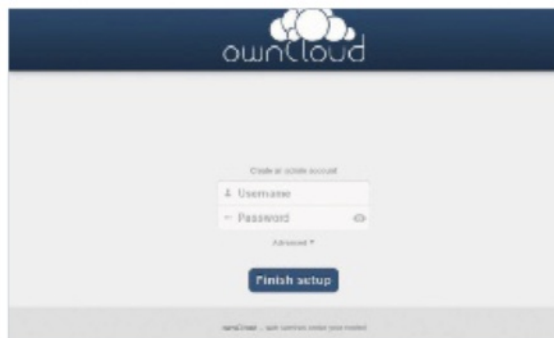
```
Begin by installing its dependencies: yum
install php-ldap php-mysql php-gd
php-imap php-odbc php-pear php-xml
php-xmlrpc. Restart Nginx and PHP-FPM:
#service nginx restart
#service php-fpm restart
```

Pay a visit to the Feng Office website (www.fengoffice.com) and download the Community Edition, supplied as a zip file. Decompress it in the web server directory with: `unzip fengoffice_2.3.zip -d /usr/share/nginx/html/feng/`. Now `cd` to that directory and type `chmod 777 config tmp upload cache` to give full access to those directories. We then create a MySQL database for Feng to use, just like we did for WordPress. Type `mysql -uroot -p` to begin.

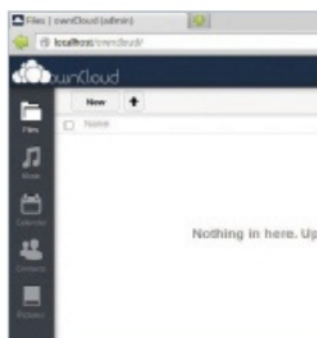
```
mysql> create database fengoffice;
mysql> GRANT ALL ON fengoffice.* TO
```

TIP Gedit

To launch a GUI text editor such as gedit while root under CentOS, use `sudo gedit [filename] &`. This sorts out the environment variables for you and detaches the terminal.



■ Logging into ownCloud for the first time



■ The ownCloud file browser

Set up port forwarding

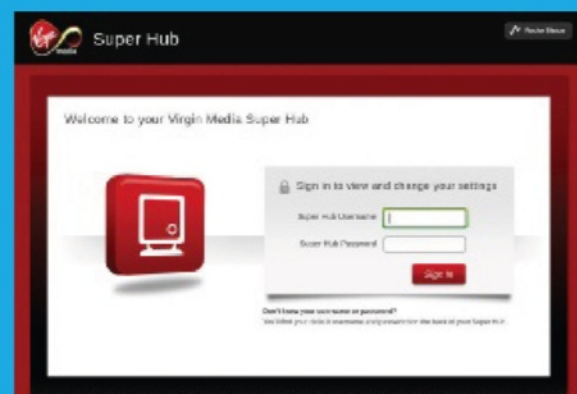
In order to enable machines outside of your LAN to connect to your server, you need to enable port forwarding. How you go about this varies depending on your make and model of router. If unsure of how to do it, visiting portforward.com is a good start because it has router-specific instructions. At the same time, use the router setup pages to discover your external IP address. Currently, searching for 'IP address' in Google will also display your current external IP address.

You can discover the IP address of your server by typing 'ifconfig' into it. You then follow the instructions specific to your router (usually accessible by using the browser to visit the first IP address on your network such as 192.168.0.1 or 192.168.1.1). You must tell it to forward ports 80 and 443 to the IP address of your server. Then, you should be able

to access the server from outside of your network by pointing a browser at the external IP address of your router.

To save having to remember this IP address, you could buy a domain name and point it at your network. However, if your ISP does not offer static

IP for customers, a dynamic DNS server might be a good alternative. Organisations such as No-IP (www.noip.com) and (www.opendns.com) offer a free subdomain that can be updated via a web interface each time you disconnect from the internet and reconnect.



■ Logging into the router setup page

```
fenguser@localhost IDENTIFIED BY
'your password';
mysql> flush privileges;
mysql> quit;
```

In a web browser, go to `localhost/feng/`. This should bring up Step 1 of the installation sequence. Clicking Next takes you to Step 2, which carries out a check of the server environment. Presuming that everything looks okay, click Next and proceed to Step 3. Here, you must enter the username, password and database name of the MySQL database that you just created. Clicking Next should allow you to log into your new Feng system.

TIP Use tab completion

Don't be intimidated by those long pathnames and filenames, use tab completion. It's faster and often guards against mistakes.

OwnCloud

OwnCloud is primarily a service for personalised file sharing. The idea is that you can access your files from anywhere and from any device, while also having the option of selectively sharing files with other people. Download the current version of ownCloud from owncloud.org. This normally comes supplied as a .tar.bz file, so the switches to decompress are slightly different. Type `tar -jxf [archive name] -D /usr/share/`

What is PHP-FPM?

PHP-FPM is a FastCGI implementation. CGI is the method that web servers use to call an executable on the host. These executables are typically written in a scripting language that the host understands, such as Perl or PHP. Traditionally, web servers opened each executable in its own process, an approach which aided simplicity. The problem is that opening and then closing processes with each request made upon the server doesn't scale very well. For this reason, FastCGI servers such as PHP-FPM use a single process to handle all CGI requests.

nginx/html, and move to that directory by typing `cd /usr/share/nginx/html`. Then, change the ownership attributes of the directory by typing `chown -R nginx owncloud`.

Create SSL certificates

We're now going to create the SSL certificates that ownCloud needs. Type `cd /etc/nginx/` to move into the configuration directory. Create a directory for the certificates: `mkdir certs`. Now move into that directory: `cd certs`. Type `openssl genrsa -des3 -out server.key 1024` to generate a 1024-bit RSA private key. This command will prompt you for a password. Whatever you choose, please make a note of it. Now type `openssl req -new -key server.key -out server.csr`. This will ask you some identifying questions, but you can hit Return to accept the defaults. Copy the key with `cp server.key server.key.orig`. Convert the key to the format we need by typing `openssl rsa -in server.key.orig -out server.key`. Sign the certificate by typing `openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt`.

Configure Nginx for ownCloud

We'll fetch an example configuration block from the official documentation. Go to the OwnCloud documentation site (doc.owncloud.org). Select Administrators Manual from the side menu. Now enter the term 'nginx' into the search box. One of the search results is the 'Other Web Servers' page and this is what we need. This page contains an Nginx example file. Of the four sections, we just need the SSL section and the WebDAV bits, disregarding the other two. Either cut and paste this into your own Nginx file (or you could create another .conf file to add



■ Installing Nginx using YUM

to your `/etc/nginx/conf.d/` if you prefer). Adapt this to your setup by altering the root string to match our web root (`/usr/share/nginx/html`). Alter the 'ssl_certificate' and 'ssl_certificate_key' lines so that they match these: `ssl_certificate /etc/nginx/certs/server.crt;` `ssl_certificate_key /etc/nginx/certs/server.key;`

OwnCloud requires multibyte string support in PHP, so install it by typing `yum install php-mbstring`. Make a final check of the current official documentation on the website to see if there are any further amendments that need

TIP LAMP or LEMP?
The most common web hosting setup on Linux is called a LAMP (Linux Apache MySQL PHP), but an Nginx setup is called a LEMP due to the phonetic sound of the name.

to be made. Restart PHP-FPM with `service php-fpm restart`.

Create ownCloud database

Create a database and a database user with a password, as with the previous examples. `CREATE DATABASE owncloud;` `GRANT ALL PRIVILEGES ON owncloud.* TO 'ownuser'@'localhost' IDENTIFIED BY 'your password';` `quit;`

Complete installation

Now browse to `localhost/owncloud` and you should be able to see the ownCloud login page. Create an administrator by specifying a username and password. Click the Advanced tab and specify your database details.

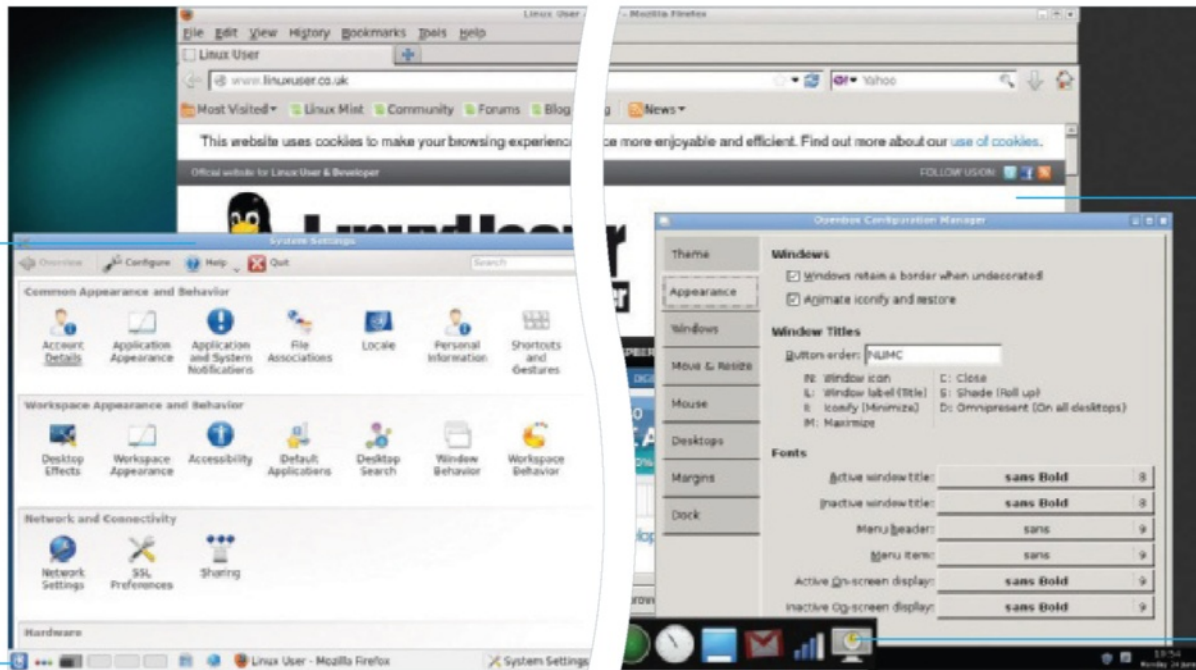
“Of the four sections, we just need the SSL and the WebDAV bits”



■ Testing that the Nginx installation has worked by pointing the browser at localhost

Speed up a core part of your system by using Openbox instead of the standard window manager

Use Openbox as a desktop environment for super-speedy and customisable workflow



Maintain the other aspects of your favourite desktop environment while using the lighter, faster Openbox

Personalise Openbox to make it more usable in the way you wish with docks, taskbars and extra menus

Speed up Linux with Openbox

Learn how to install and properly configure the lightweight window manager Openbox and speed up your day-to-day computing without sacrificing usability

We're always looking for ways to speed up our systems. Whether we're trying out lighter distros or desktop environments, building from scratch or selecting the perfect array of apps, there are many ways to accomplish this. One of the ways that can sometimes get overlooked, though, is changing your window manager – the set of packages that handles the actual windows of your desktop.

One of the most popular and lighter window managers is Openbox. It's one of the main window managers in LXDE, and readily available to a lot of distros either through their repos or

via the website. It can noticeably speed up your desktop, especially if you're using GNOME, KDE and the like.

Openbox can also be used as your main, supercharged and minimal desktop environment. It uses a much simpler layout than some of the more popular desktop environments; however, it's perfectly usable with a few tweaks and may just greatly increase your workflow.

The best part is, you can go back easily to your old desktop or windows manager whenever, thanks to the way Linux login managers handle desktop sessions.

Resources

Openbox:
openbox.org/wiki/Openbox:Download



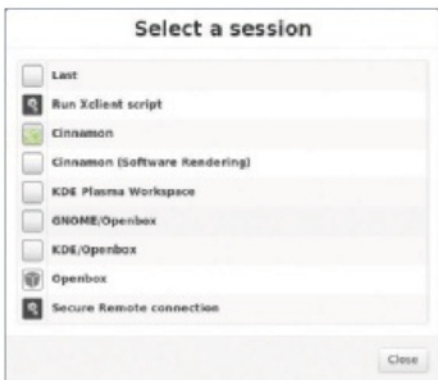
01 Install Openbox

Installing Openbox to your system is very easy. You can find it in your graphical package manager or software centre, or install it with the following for Debian-based systems:

```
$ sudo apt-get install openbox
```

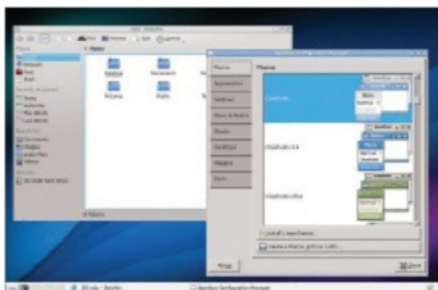
...and for Fedora it's:

```
$ sudo yum install openbox
```



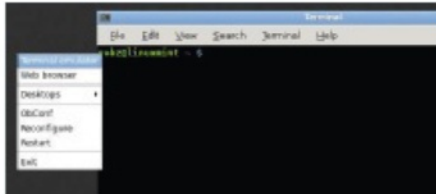
02 Use Openbox

Log out of your distro. MDM, GDM, LightDM and KDM will all allow you to select a session at the login screen – open the selection and you'll see that you now have the option to use GNOME/Openbox or KDE/Openbox.



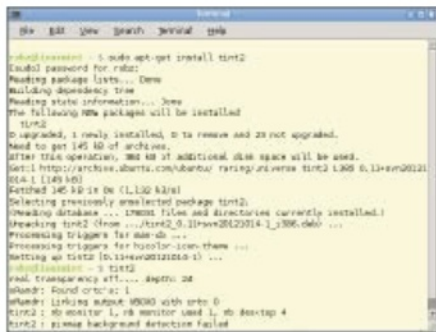
03 Basic configure

Openbox is highly configurable, and the most basic configuration can be found in the graphical manager for this. Here you can change the windows theme, the way the windows react during your workflow, and whether or not you want to use a dock.



04 Log into Openbox

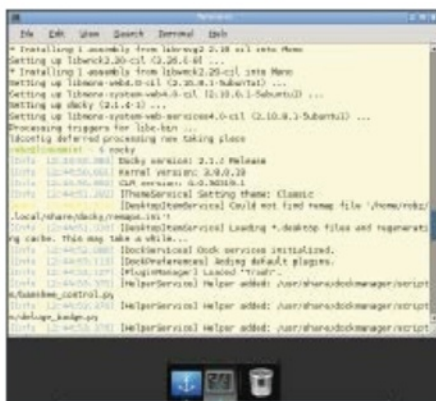
Log back out and select the Openbox desktop from your session manager. After logging back in, you'll be presented with a basic grey desktop and not much more. Right-clicking will open up some options; for now, open the terminal.



05 System tray

To get a panel with open windows and system trays, your best bet is to install tint2. To do this, you'll need to simply install it using the terminal we just opened. The package is called tint2, so for Fedora it would be:

```
$ sudo yum install tint2
```



06 Docking

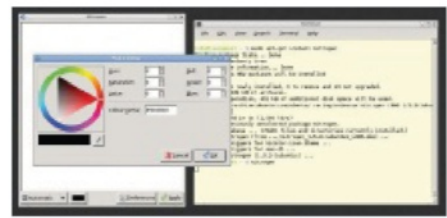
You can create a shortcut dock for apps to live on, similar (but better) than what you get in OS X. It was used in Fuduntu, and is nice and lightweight. To install Docky in something like Debian, use:

```
$ sudo apt-get install docky
```

07 Desktop compositing

For Docky to work properly, there needs to be some degree of desktop compositing. One of the best ways to do this while still keeping a quick system is to use xcompmgr. Install on Fedora with:

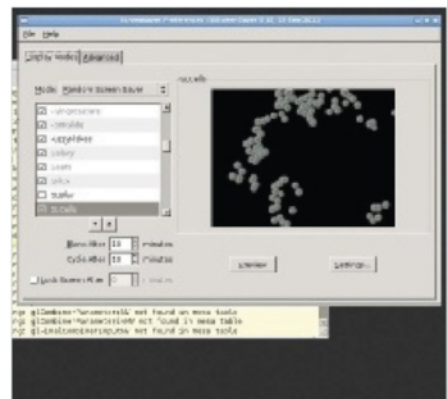
```
$ sudo yum install xcompmgr
```



08 Backgrounds

To be able to set a background image and fully customise your Openbox desktop, the best package for the job is Nitrogen. It comes with a graphical interface to choose backgrounds and can be installed with:

```
$ sudo apt-get install nitrogen
```



09 Saving screens

You can install a screensaver to Openbox by using the basic xscreensaver. Install it with something like:

```
$ sudo yum install xscreensaver  
xscreensaver-g1
```

To modify it, run `xscreensaver-demo` from the terminal. This also adds power management options.

10 Autostarting

These will not automatically start when logging into Openbox, so we need to create an autostart script to deal with it. Create a config directory with:

```
$ mkdir ~/.config/openbox
```

...and then open a new autostart file with:

```
$ nano ~/.config/openbox/autostart
```



11 Start script

Add the individual elements to the autostart script like so:

```

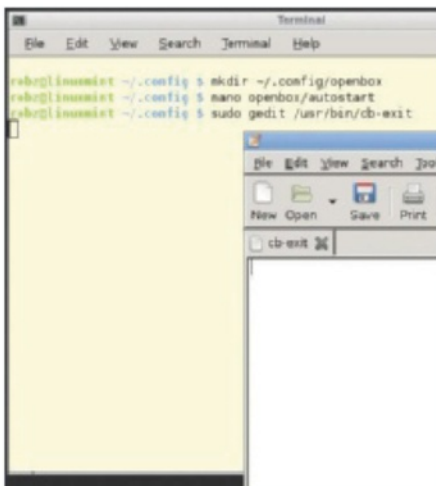
nitrogen --restore &
tint2 &
xcompmgr -c -t-5 -l-5 -r4.2 -o.55 &
docky &
xscreensaver -no-splash &
    
```

Press Ctrl+X and save the script.

12 Numlock on

By default, the numlock will not be kept on when logging into Openbox. To get this to happen at startup, install the numlock x package with yum or apt-get, and then add this line to the autostart script:

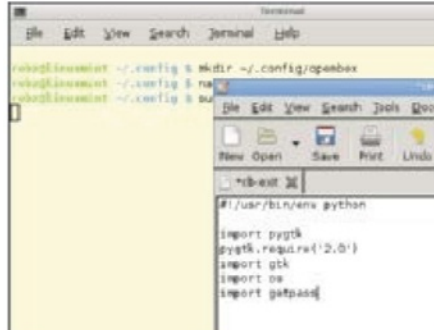
```
numlockx on &
```



13 Shut down

Openbox doesn't have a specific menu that lets you shut down graphically. Crunchbang, a Linux distro that uses Openbox, has a great Python script for this that we can borrow from. First of all, create the shutdown menu script with:

```
$ sudo gedit /usr/bin/cb-exi
```



14 Python imports

Set up the script so we can use the necessary Python elements with:

```
#!/usr/bin/env python
```

```

import pygtk
pygtk.require('2.0')
import gtk
import os
import getpass
    
```

15 Shutdown confirmed

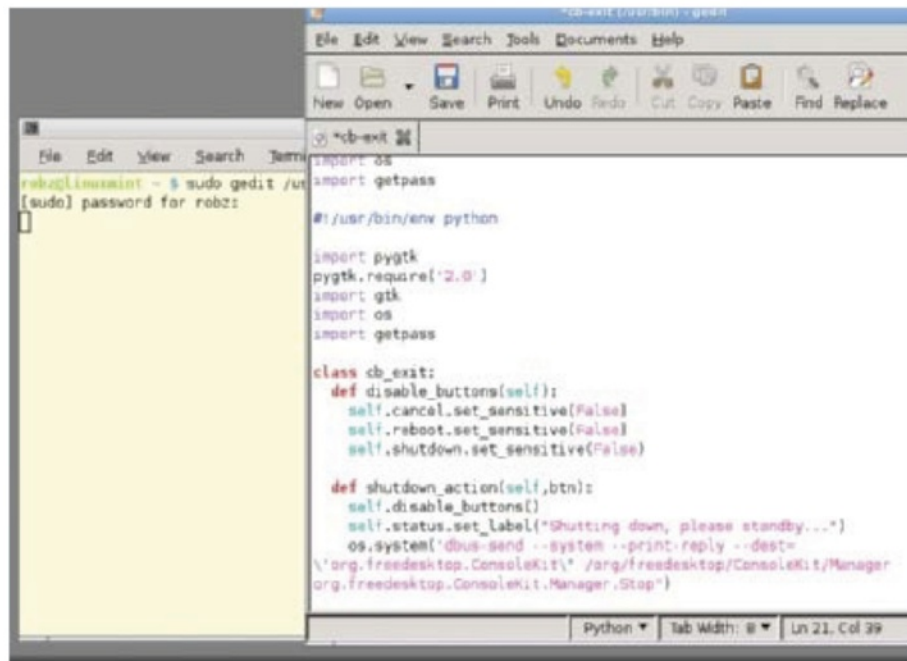
For a simple shutdown button, you'll need to do the following in the script:

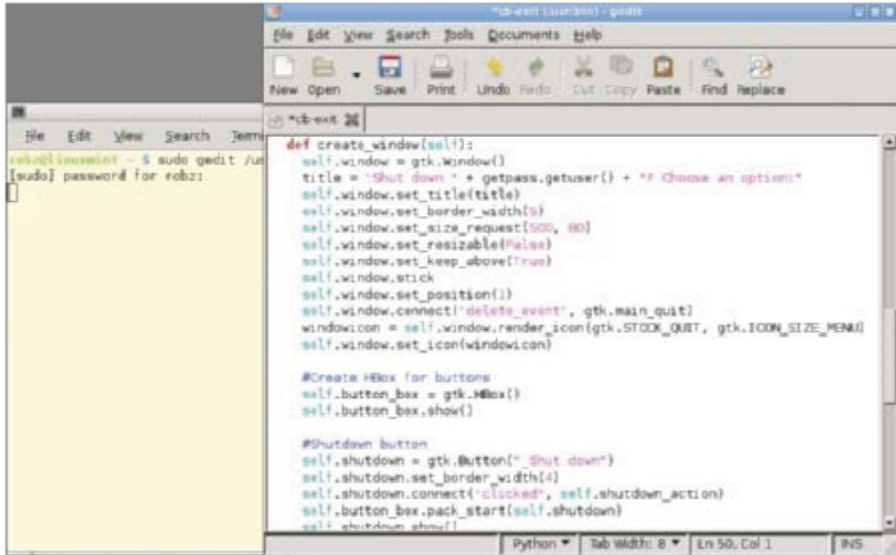
```

class cb_exit:
    def disable_buttons(self):
        self.shutdown.set_
sensitive(False)
    def shutdown_action(self,btn):
    
```

```

        self.disable_buttons()
        self.status.set_
label("Shutting down, please
standby...")
        os.system("dbus-send
--system --print-reply --dest=\\"org.
freedesktop.ConsoleKit\\" /org/
freedesktop.ConsoleKit/Manager org.
freedesktop.ConsoleKit.Manager.Stop")
    def create_window(self):
        self.window = gtk.Window()
        title = "Shut down " +
getpass.getuser() + "? Choose an
option:"
        self.window.set_title(title)
        self.window.set_border_
width(5)
        self.window.set_size_
request(500, 80)
        self.window.set_
resizable(False)
        self.window.set_keep_
above(True)
        self.window.stick
        self.window.set_position(1)
        self.window.connect("delete_
event", gtk.main_quit)
        windowicon = self.window.
render_icon(gtk.STOCK_QUIT, gtk.
ICON_SIZE_MENU)
        self.window.set_
icon(windowicon)
    
```





16 Shutdown button

That's the function of the button set up; now for the window and button:

```
self.button_box = gtk.HBox()
self.button_box.show()
self.shutdown = gtk.Button("~
Shut down")
self.shutdown.set_border_
width(4)
self.shutdown.
connect("clicked", self.shutdown_
action)
self.button_box.pack_
start(self.shutdown)
self.shutdown.show()
```

```
self.vbox = gtk.VBox()
self.vbox.pack_start(self.
button_box)
self.vbox.pack_start(self.
label_box)
self.vbox.show()
self.window.add(self.vbox)
self.window.show()
def __init__(self):
    self.create_window()
def main():
    gtk.main()
if __name__ == "__main__":
    go = cb_exit()
    main()
```

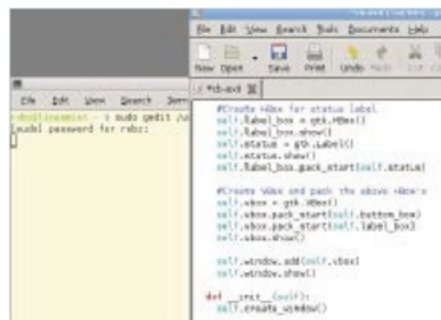
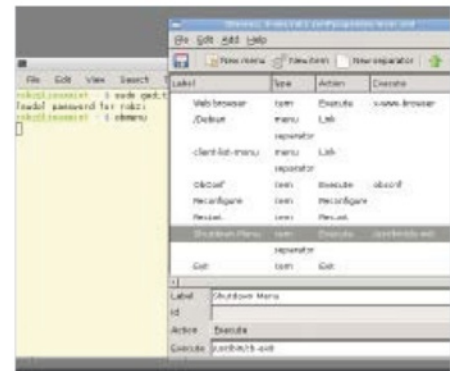
```
def cancel_action(self, btn):
    self.disable_buttons()
    gtk.main_quit()
```

And then adding the button with:
**self.cancel = gtk.Button(stock =
 gtk.STOCK_CANCEL)**
self.cancel.set_border_width(4)
**self.cancel.connect("clicked", self.
 cancel_action)**
self.button_box.pack_start(self.cancel)
self.cancel.show()

20 Menu button

To add this shutdown menu to the Openbox menu, you'll need to install obmenu. This graphical tool can help you add apps and scripts to the menu, and is easy to use and very customisable. Install it with something like:

```
$ sudo yum install obmenu
```



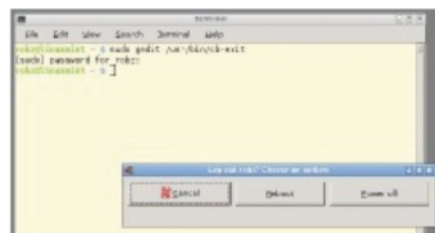
17 Shutdown window

Finally, we finish off the script like so:

```
self.label_box = gtk.HBox()
self.label_box.show()
self.status = gtk.Label()
self.status.show()
self.label_box.pack_
start(self.status)
```

18 Restart button

To add a reboot button involves almost the same code as the shutdown button. While defining reboot_action, make it the same as shutdown_action, but make sure to use the .Restart function from the ConsoleKit. Create the reboot button by simply replacing 'shutdown' with 'reboot' in the same code.



19 Cancel button

You can add a cancel button by defining cancel_action like so:

21 Add button

Open obmenu, and expand the Openbox 3 arrow. Choose a place to add the button and press New Item. Give it any label you wish, such as Shutdown, make sure Action is set to Execute, and set the Execute command to /usr/bin/cb-exit.



22 Extra menus

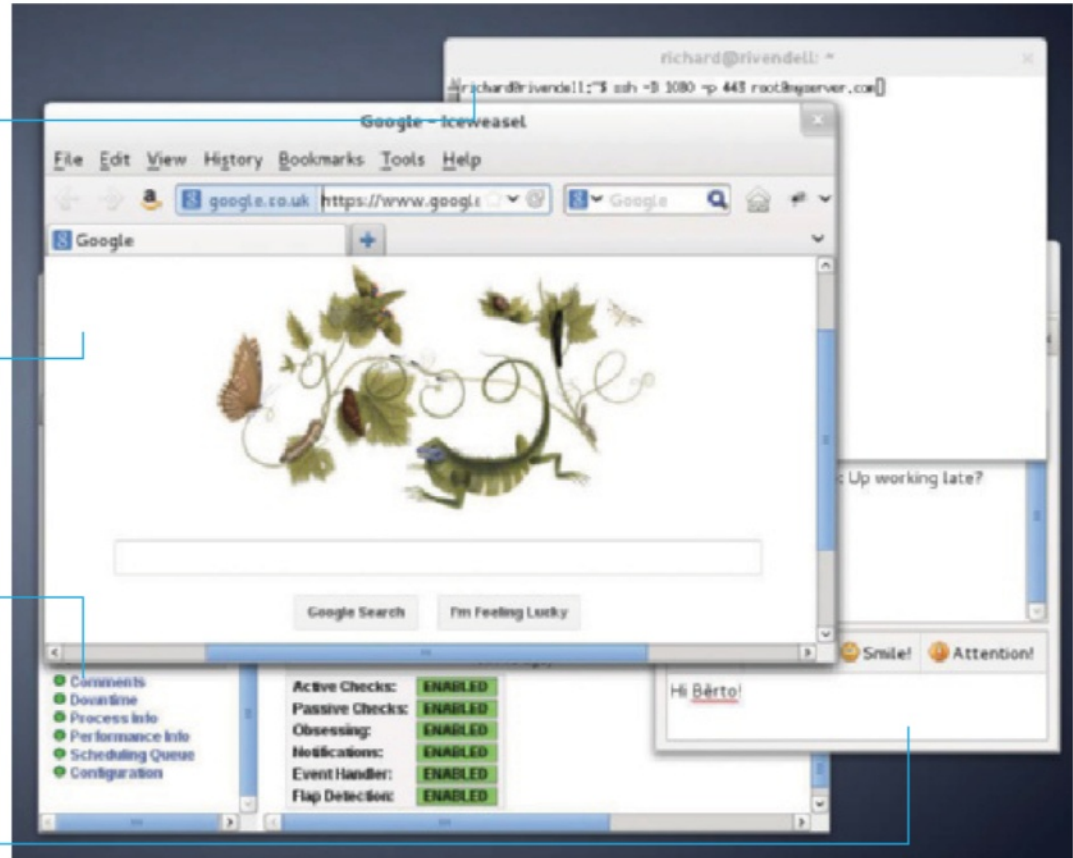
From here you can add extra buttons, apps and functions to the right-click menu and customise your experience. There's a lot of extra customisation you can do with Openbox in general as well, with theming options, behaviour options and much more.

Create a SOCKS proxy server and get your services past firewalls that block the necessary ports

Bypass over-restrictive content filtering by tunnelling your browsing via your server, protecting your unencrypted web traffic from insecure networks, too

Monitor your servers from outside the network without running the corporate VPN client – connect on the fly from any device for out-of-hours monitoring

Need to chat to coders or your office on Jabber when the client site's firewall blocks the XMPP port? Tunnel it over an open port and simply connect your chat client to localhost:8080, for example



Bypass restrictive firewalls using SSH tunnelling

Create secure network connections on the fly and run safely over insecure networks

If you're still using SSH as just a telnet replacement, you are missing out on borrowing its secure encryption to carry many other network services through insecure Wi-Fi, and overly restrictive firewalls, from wherever you have a laptop or smartphone.

For the bulk of this article, we shall be looking at local port forwarding – the most common and the most useful type – to give secure, VPN-like connections. Why not just use a virtual private network? VPNs aren't always available to you, and some corporate VPNs demand particular client software and configuration, but SSH

tunnels can always be created on the fly, as and when you need them.

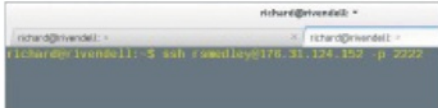
Perhaps you have never read the SSH man page? No? Well, the options you should have been looking at are `-L` and `-R`, with a little attention to `-N` and `-f`.

Skipping lots of theory, we'll take a practical approach and show you how to use SSH tunnelling in various common scenarios. Read on and find just what these magic switches to the `ssh` command can do for you, but beware – the power to run rings around firewalls should be used carefully!

Resources

SSH client with SSH daemon on the server

A server connected to the internet, preferably with a fixed IP address



01 A different port

When you run a normal SSH session, it simply opens an encrypted connection from a spare port on your computer to port 22 on a remote device. For security reasons – many scripts are knocking on port 22 with well-known passwords – you can specify another port.



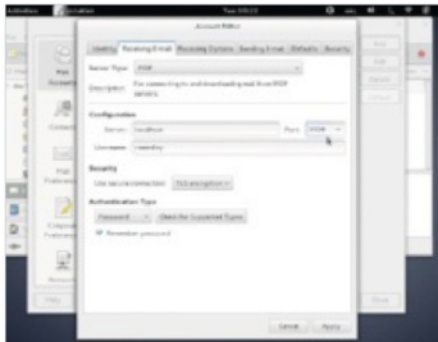
02 Insecure access

However, inside this encrypted connection you can carry other traffic – hence SSH tunnelling. This means that however insecure your connection (eg cafe Wi-Fi), your traffic is as secure as the level of encryption used by SSH (ie good enough).



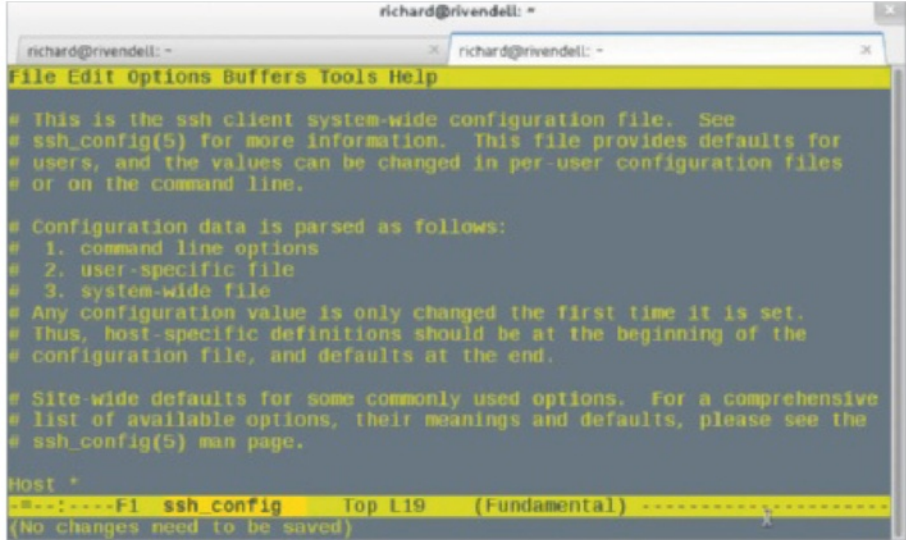
03 Confidential mail

Tunnelling allows you to hide your unencrypted email traffic inside the SSH connection. The -L local-port:host:remote-port creates the tunnel, allowing SMTP (port 25) traffic from the mail-server to appear on (for example) port 3909 locally.



04 Local config

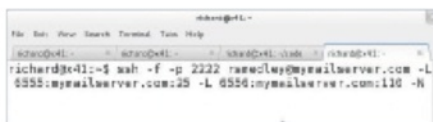
Now just configure your email client to connect to port 3909 of the local machine. Localhost and 127.0.0.1 are synonymous, but you



could also use the fully qualified domain name (FQDN) of your local machine. You can do the same for receiving mail via POP.

05 Pick a number

Why port 3909? Port numbers below 1024 are for privileged services. No non-root users should be looking higher than this, but taking a peek at the popular ports in use by other software. Pick a free number such as 6555 or 3989 as your default.

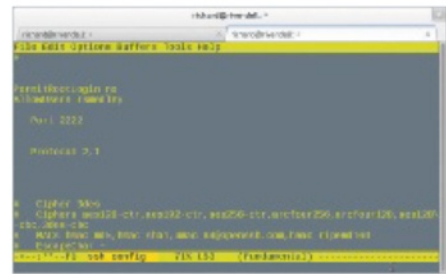


06 Two-lane tunnel

While outward-bound SMTP is occasionally blocked, if you're tunnelling for security, best do the incoming POP mail with the same command. As you can see, multiple local tunnels can be expressed in the same ssh command, each with the -L switch.

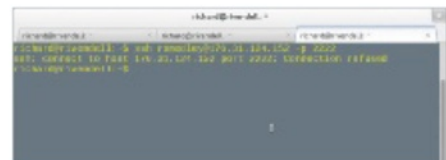
07 On the server

Before we go any further, best get a couple of things straight on our server. SSH in (without the tunnel this time), gain root privileges, and fire up your favourite editor to open /etc/ssh/sshd_config (or whatever your distro names the file).



08 Security basics

As well as security settings like a port other than 22, and not allowing root login, here you should uncomment the protocol version 2 setting, so only the more secure protocol version 2 will be used. If both are listed, delete the '1'.

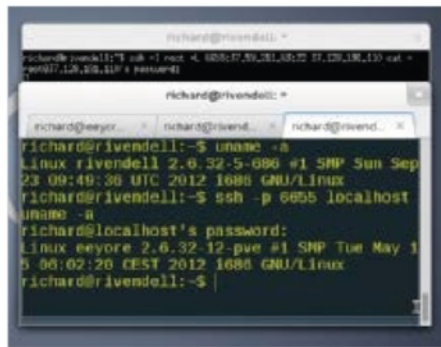


09 Error check

Check you can log in on the new port from another terminal before you close this session! If there is a problem, check that you restarted the SSH server, and typed the correct port and username. If in doubt, return to default port setting.

outside world:

```
ssh -l username -L 6655:hiddenmachine:22 gatewayserver cat -
```



17 SSH over SSH

Now we can SSH to the chosen port (6655) on localhost, and we will be executing commands on the hidden server. You can also execute slogin, SCP or SFTP via localhost, port 6655 – tunnelling right through the gateway machine (visible server).



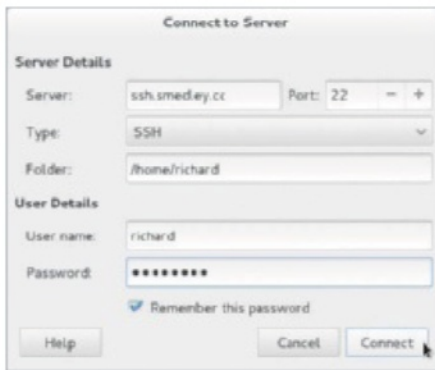
18 Power of reverse

A reverse tunnel lets you connect to a NATed machine, without a public IP address. The NATed machine opens a reverse tunnel to a server, and from the server one opens a connection to localhost and the chosen port which connects you back down the tunnel.



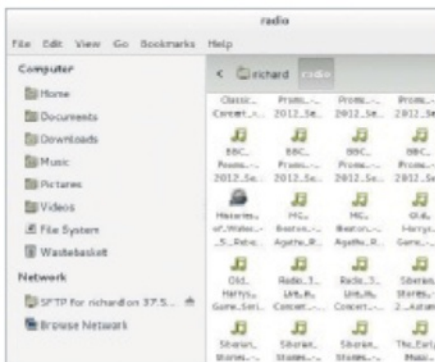
19 Third-party access

From a third machine, connect to the server. Then SSH to localhost and you are also connected to the NATed machine. This means from anywhere you can connect to a desktop without an SSH server, if it can run a client.



20 GUI help

Some desktop software effectively tunnels through SSH for you, such as your file browser. In Nautilus, go to **File --> Connect to Server** and put in your SSH details. In Konqueror enter **fish://user@server** in the location bar.



21 Drag and drop

Now you can work on remote sites alongside your desktop and locally mounted shares. Who needs Dropbox? Note that as well as SSH, you can do this over FTP or HTTP (WebDAV). GUI-haters can use MC (from the Right menu, select 'shell link'), or mount with SSHFS.



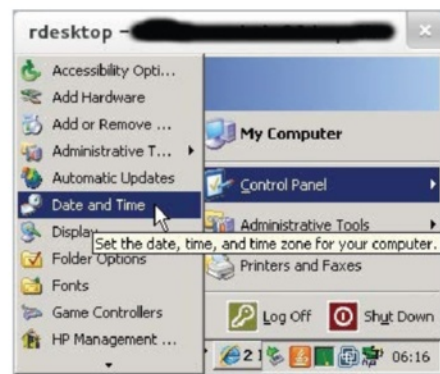
22 Remote apps

At its simplest, tunnelling X applications means never having to battle dependencies to install difficult apps on your PC, so long as they're running on a machine to which you have SSH access with X forwarding enabled. In practice, machines on local networks give best (least laggy) results.



23 Transcontinental apps

Nevertheless, graphical apps can be run from servers hosted in another country, as long as you are prepared to put up with a little lag in busier apps. You could even browse BBC iPlayer on a UK-hosted box while travelling overseas.



24 Remote desktop

Beyond forwarding Z apps, we'll have a bit more to say on VNC and remote desktops next month, when we conclude our look at secure remote network apps and look at the more permanent alternative to SSH tunnels – the virtual private network or VPN.

Make a decision from a small selection of ISOs to make the build easier for your environment



Customise your system down to the smallest part to make sure it runs as smooth as possible

Customise the kernel so that there's no bloat in your system



Create a custom build of Gentoo

Build a custom distro from the ground up to suit your preferences and even speed up your system

Gentoo can be a double-edged sword. It's probably the most customisable Linux distribution available, letting you build it from the ground up to be exactly what you want it to be. However, it's not all that easy to get to grips with and requires some serious skills in Linux to get right.

Once you have, though, Gentoo can be very rewarding. Building packages from source and compiling the kernel yourself feels like you're making the most out of your Linux experience, and can actually help make your system a

lot faster than some of the more popular, preconfigured distros.

In this tutorial, we'll cover taking an image and doing the first-time setup. While we'll be relying on the minimal installation disc and downloading stage tarballs from the internet, there is also a more complete DVD image that you can use to create a more rudimentary setup. A lot of the steps will be the same; however, if you get stuck, the Gentoo website has some great resources and manuals to guide you through any differences.

Resources

Gentoo live image:
www.gentoo.org/main/en/mirrors2.xml

Name	Last
Parent Directory	
hardened/	29-M
install-x86-minimal-20130528.iso	28-M
install-x86-minimal-20130528.iso CONTENTS	28-M
install-x86-minimal-20130528.iso DIGESTS	28-M
install-x86-minimal-20130528.iso DIGESTS.asc	29-M
stage3-i486-20130528.tar.bz2	28-M
stage3-i486-20130528.tar.bz2 CONTENTS	28-M
stage3-i486-20130528.tar.bz2 DIGESTS	28-M
stage3-i486-20130528.tar.bz2 DIGESTS.asc	29-M
stage3-i686-20130528.tar.bz2	28-M
stage3-i686-20130528.tar.bz2 CONTENTS	28-M
stage3-i686-20130528.tar.bz2 DIGESTS	28-M
stage3-i686-20130528.tar.bz2 DIGESTS.asc	29-M

01 Install and boot

Grab the minimal Gentoo live image and install it to a CD in whatever way you prefer. Restart your system and boot from CD/DVD, and press Enter at the boot screen. You'll then be asked if you want to change your keymap with some option, otherwise it will take you to the live system command line.



02 Partitioning

We now need to set up the disc so we can install to it. We'll use Parted. First, check the current layout with:

```
# parted /dev/sda
```

And then enter `print` in the new Parted command prompt. Use `rm [number]` to remove partitions of that number that you don't need.

03 Create partitions

We're going to start from a blank hard drive. We'll need a boot partition, a swap and some space. Use the following three commands:

```
mkpart primary ext2 0 32mb
```

```
mkpart primary linux-swap 32mb [32 + RAM]mb
```

```
mkpart primary ext4 [32 + RAM]mb -1s
```

Agree to or ignore any prompts. The option at the end of the third command tells Parted to fill up the rest of the disc.



04 Make file systems

Quit out of Parted with `quit`. You'll now need to make the partitions into file systems using the following commands:

```
# mkfs.ext2 /dev/sda1
# mkfs.ext4 /dev/sda3
```

```
Create the swap with:
# mkswap /dev/sda2
```

And then swapon with:

```
# swapon /dev/sda2
```

05 Mount partitions

Before we continue, we need to mount the partitions. Do this by first mounting the storage as `/mnt/gentoo`:

```
# mount /dev/sda3 /mnt/gentoo
```

Create a boot folder within this:

```
# mkdir /mnt/gentoo/boot
```

Mount the boot folder:

```
# mount /dev/sda1 /mnt/gentoo/boot
```

06 On time

Make sure the clock is correctly configured to UTC time by simply entering `date`. If it's not, make it UTC time with the following command:



```
# date MMDDhhmmYYYY
```

...where MM is the month, DD the day/date, etc.

07 Take the stage

Move to mount point you just created with `cd`, and then type the following to get a list of mirrors for the stage3 tarball:

```
# links http://www.gentoo.org/main/en/mirrors.xml
```

Navigate using the arrow keys to your nearest mirror, go to releases, then your architecture, current stage3, and download a stage3 tarball.

08 Extraction

Unpack the tarball you just downloaded with this:

```
# tar xvjpf stage3-*.tar.bz2
```

Once it's unpacked, open the configuration file using nano:

```
# nano -w /mnt/gentoo/etc/portage/make.conf
```

And then we will be able to start some of the initial configuration.

09 Make options

The default options already in the configuration file should be good enough for most systems. We can also add an option for how many parallel compilations can occur at once by adding this to the bottom:

```
MAKEOPTS="-j[X]"
```

...where X is the number of cores you have plus one. Save and exit with `Ctrl+X`.



10 Prepare build environment
We're nearly ready to start building. Save the network/DNS details to the environment with:

```
# cp -L /etc/resolv.conf /mnt/gentoo/etc/
```

Next, mount the `/proc` file systems, and then bind them to `/dev` and `/sys` with:

```
# mount -t proc none /mnt/gentoo/proc
# mount --rbind /sys /mnt/gentoo/sys
# mount --rbind /dev /mnt/gentoo/dev
```



11 Enter build environment
The build environment is technically your new Linux environment. We need to make some changes so we can enter it first – basically change the directory we created to root using `chroot` with:

```
# chroot /mnt/gentoo /bin/bash
# source /etc/profile
# export PS1="(chroot) $PS1"
```



12 Portage
We need the latest Portage snapshot before we go any further. Emerge it with:

```
# emerge-webrsync
```

This will allow us to install all the packages we need. Update Portage before continuing with:

```
# emerge --sync
```

13 Installation profile
You'll now need to choose how to build Gentoo by setting a specific profile. There are three profiles, two of which are of interest to us – Desktop and Server. This will determine the type of packages we use. Set it with:

```
# eselect profile set 2
```

2 is desktop; change it to 3 for a server.

14 USE me
The USE variable in `make.conf` is a powerful tool to configure compiling so it only installs the package support you require. A full list of these flags can be found online or in your system at:

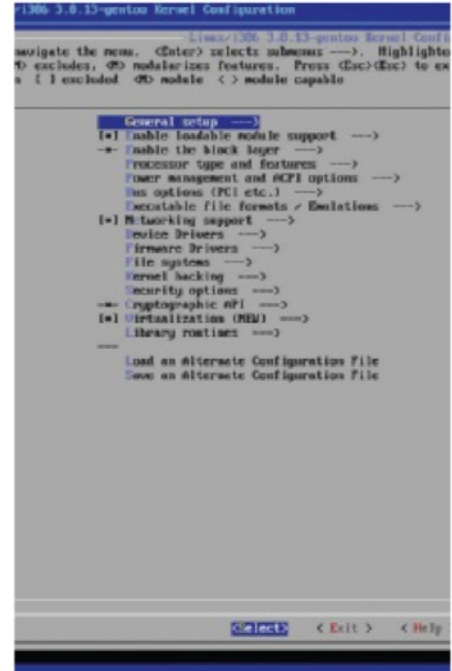
```
# less /usr/portage/profiles/use.desc
```

We'll make our system so it will install files for GNOME and GTK, as well as add ALSA and DVD support. Edit the make file with:

```
# nano -w /etc/portage/make.conf
```

And change USE to:

```
USE="gtk gnome -qt4 -kde dvd alsa"
```



15 Kernel time
List the available time zones with:

```
# ls /usr/share/zoneinfo
```

For London, we will do the following:

```
# cp /usr/share/zoneinfo/Europe/London /etc/localtime
```

```
# echo "Europe/London" > /etc/timezone
```

Now it's time to download our kernel. First, get `gentoo-sources` with:

```
# emerge gentoo-sources
```

Check what kernel version `gentoo-sources` is pointed at with:

```
# ls -l /usr/src/linux
```

From here, you can start modifying the kernel flags by entering:

```
# cd /usr/src/linux
# make menuconfig
```

Make sure to change only the kernel options you need to. Activate any other required modules. Once done, exit the configuration.

16 Compile kernel

The moment of truth – time to compile your kernel. Do this with:

```
# make && make modules_install
```

This will take a while depending on the amount of modules and options you activated. Once it's finished, install the kernel with:

```
# cp arch/x86/boot/bzImage /boot/  
kernel-[X]-gentoo
```

...with X being the number we found last step.

17 Boot modules

You'll need to set what kernel modules you want to load. To find what modules are available, use:

```
# find /lib/modules/[X]/ -type f  
-iname '*.o' -or -iname '*.ko' |  
less
```

...again where X is the kernel number. You then need to add the modules you want to this file:

```
# nano -w /etc/conf.d/modules
```

18 Tabbed file system

We need to set the partitions we created to be mounted properly at boot. Open fstab with:

```
# nano -w /etc/fstab
```

And then set the following options so that the file system we set up works properly:

```
/dev/sda1 /boot ext2 defaults,noatime 0 2  
/dev/sda2 none swap sw 0 0  
/dev/sda3 / ext4 noatime 0 1
```

19 Networking

You'll need to configure your network for after the reboot. First enter the config file with:

```
# nano -w /etc/conf.d/net
```

And add this like:

```
config_eth0="dhcp"
```

If you're using static IPs, you can add them instead of DHCP. Save, and then make it bootable with:

```
# cd /etc/init.d  
# ln -s net.lo net.eth0  
# rc-update add net.eth0 default
```



20 Root setup

Set the root password with the standard `passwd` command. Now set some basic services by editing:

```
# nano -w /etc/rc.conf
```

Keymaps with:

```
# nano -w /etc/conf.d/keymaps
```

And the hardware clock:

```
# nano -w /etc/conf.d/hwclock
```

If this is not UTC, add `clock="local"` to the file

21 Your locale

Specify your locales for the system. A basic setup will need you to edit:

```
# nano -w /etc/locale.gen
```

...and add:

```
en_GB ISO-8859-1  
en_GB.UTF-8 UTF-8
```

Save, exit and then type `locale-gen`. Set it as default in `/etc/env.d/02locale` with:

```
LANG="de_DE.UTF-8"  
LC_COLLATE="C"
```

And then reload the environment with:

```
# env-update && source /etc/profile
```

22 Bootloading

We need to install GRUB so we can boot into Gentoo after a restart. Compile it with:

```
# emerge grub
```

Now create the `grub.conf` file with:

“Build packages from source and compile the kernel yourself to make the most out of your Linux experience”

```
# nano -w /boot/grub/grub.conf
```

And get ready to add the necessary details.

23 GRUB code

```
default 0  
timeout 15
```

```
title Gentoo Linux  
root (hd0,0)  
kernel /boot/kernel-3.8.13-gentoo  
root=/dev/sda3
```

```
title Gentoo Linux (rescue)  
root (hd0,0)  
kernel /boot/kernel-3.8.13-gentoo  
root=/dev/sda3 init=/bin/bb
```

Make sure to use the correct kernel number. Save and exit.

24 GRUB setup and reboot

Create an `mtab` to install GRUB to with:

```
# grep -v rootfs /proc/mounts > /  
etc/mtab
```

And finally, install it with:

```
# grub-install --no-floppy /dev/sda
```

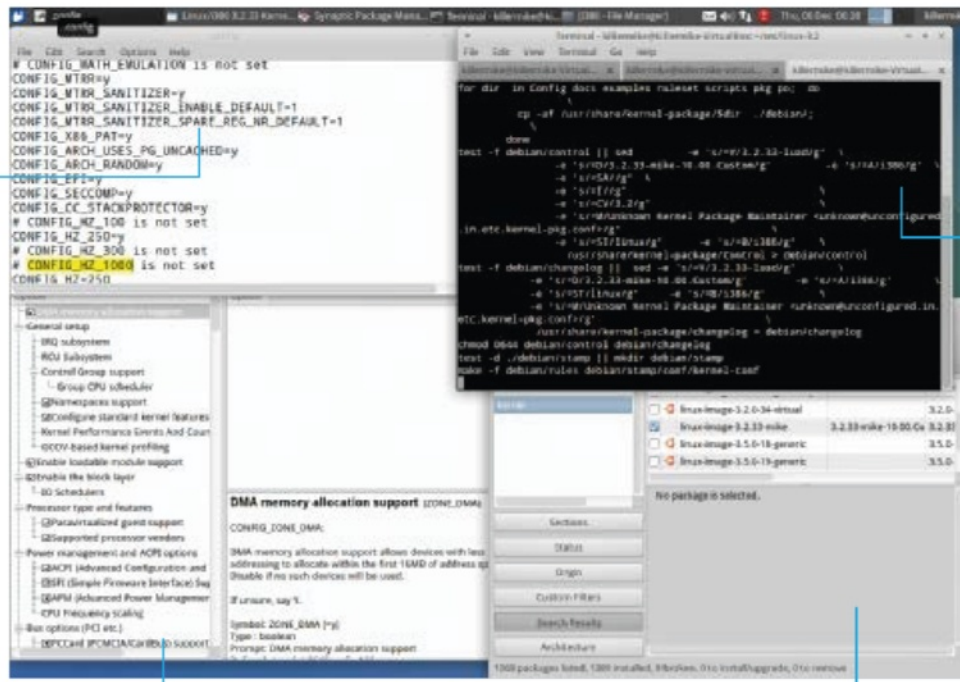
To reboot into your brand new system, exit the `chroot` and reboot with:

```
# exit  
cdimage ~# cd  
cdimage ~# mount -l /mnt/gentoo/  
dev{/shm,/pts,}  
cdimage ~# mount -l /mnt/gentoo{/  
boot,/proc,}  
cdimage ~# reboot
```

Create a custom Linux kernel to optimise performance

We'll take you through the steps you need to compile your own customised kernel for performance, specialised use and simply to learn how the plumbing works

.config is the file in which you will make most of your changes



Compiling the kernel takes a long time, but fortunately, you can leave it running in a terminal window and get on with other work

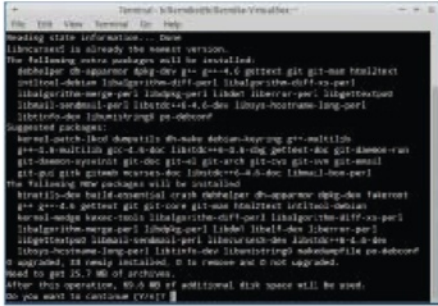
Fortunately, there are a few different ways of editing .config, including xconfig, a GUI editor

Once the kernel is complete, you can add it to the system using the standard package tools

Back in the mid-1990s, recompiling the kernel was something of a necessity, and it was also a good test that a user had mastered the basics of administering Linux. These days, the stock kernel that comes with most distros has much improved, removing the necessity of kernel recompilation for basic use. However, there are cases where it's well worth becoming familiar with this area of tweaking your system. For one thing, it's a must if you want to access the latest and greatest kernel improvements, hot off the press, so to speak. It's also a good way of understanding how the kernel and other fundamental parts of a Linux system actually

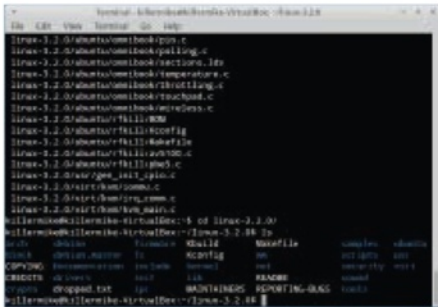
work. It can also be useful when troubleshooting: the newest kernel might bug-fix the problem you're having. On the other hand, an older kernel might be the workaround that you need.

We'll start you off with a simple example that begins with fetching the source archive for the kernel that you are currently running, proceeding through to configuration, compilation and installation. Following this, we'll go through some examples that are a bit more specialist. Most of the examples are for Debian-derived distros, but we've deliberately kept things as neutral as possible and added some notes for how to handle things on Red Hat-based distros such as Fedora.



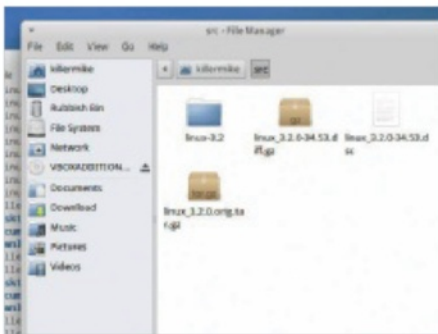
01 Install prerequisites

Begin by fetching the tools needed to create a suitable build environment. Enter 'sudo apt-get install fakeroot crash keexec-tools makedumpfile kernel-wedge git-core libncurses5 python libncurses5-dev kernel-package libelf-dev binutils-dev' followed by 'sudo apt-get build-dep linux-image.'



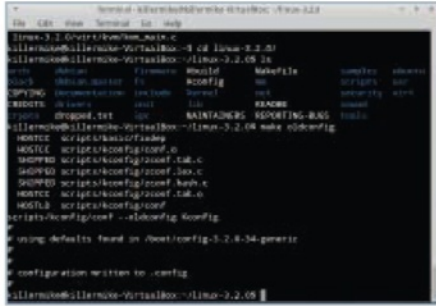
02 Fetch the kernel (source archive)

If you need the latest kernel, use Git to fetch it (see later step), but we are going to use the standard package tools in the first example. Use 'apt-get source linux-image-\$(uname -r)' to install the source for the currently running kernel.



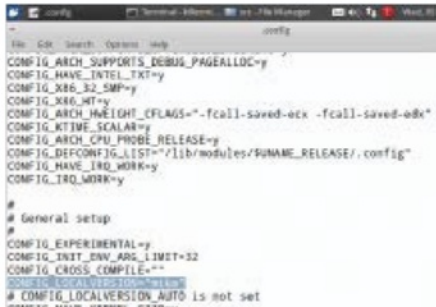
03 Examine the source directory

You should now have a source directory in the current directory. Move into it using the cd command. Note that there is an archived (tar.gz) copy as well. In addition, there is a diff file that contains the Ubuntu-specific additions to the standard kernel source tree.



04 Generate a .config file

The (hidden) file '.config', located in the source code directory, tells the compiler what to build. The configuration file for each installed kernel is stored in the /boot directory, but you can capture the configuration of the current kernel (a good starting point) by typing 'make oldconfig'.



05 Edit .config

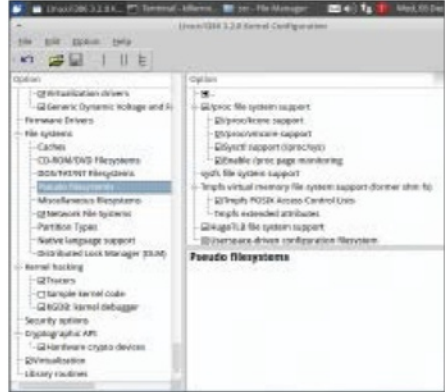
Open up .config in a text editor. Note that there are thousands of options, and this approach is best if you know exactly what settings you would like to edit. It's a good idea to search for 'CONFIG_LOCALVERSION' to add a small identifying string for your custom kernel.



06 Turn off debugging

One way to speed things up and produce smaller files is to turn off debugging.

It's a specialist feature and mainly used by developers. You can use xconfig for this. Set 'CONFIG_DEBUG_INFO:' to 'n'.

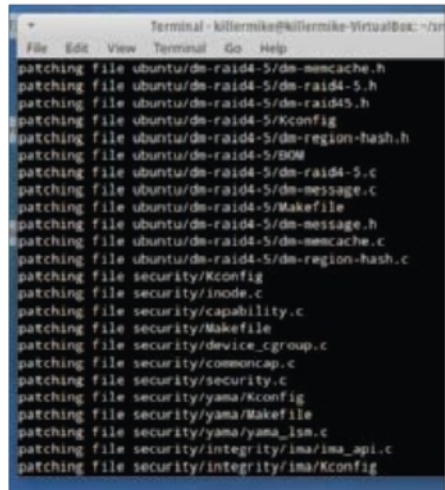


07 Invoke xconfig

Type 'make xconfig' to launch the GUI config file editor. It's a good way to gain an overview, and it offers information for most of the options. Run 'sudo apt-get install libqt4-core libqt4-dev libqt4-gui' if it complains about not being able to find Qt.

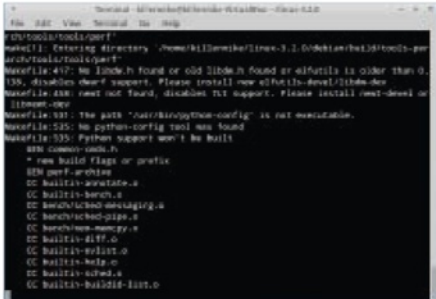
08 Prepare Debian scripts

Some required scripts lose their execution privileges due to how apt-get works. Rectify this by typing 'chmod -R u+x debian/scripts/*' and then 'chmod u+x debian/rules'.



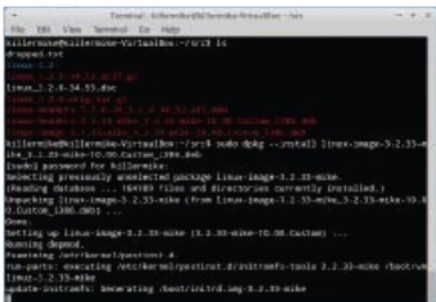
09 Recreating the Source Tree

If you mess things up and want to start from scratch, delete the source directory. Backup your .config file first, if needed. Then run "tar xzf" on the source archive to unpack it. Move into the directory and type "zcat ./[name of diff archive] | patch -p1" to add the Ubuntu patches into the source tree.



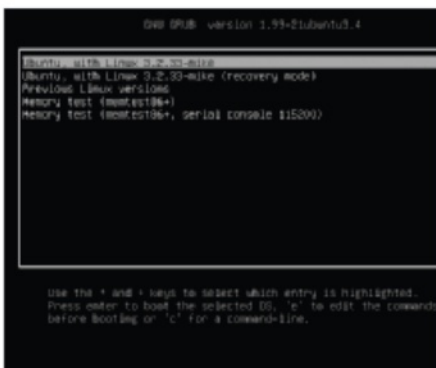
10 Compile the kernel

Tell the build environment how many cores you want to use with 'export CONCURRENCY_LEVEL=[number of cores]'. Begin the build process with 'fakeroot make-kpkg --initrd --append-to-version=-luad kernel-image kernel-headers', which takes about an hour on a Core Duo 2.7GHz system.



11 Install kernel

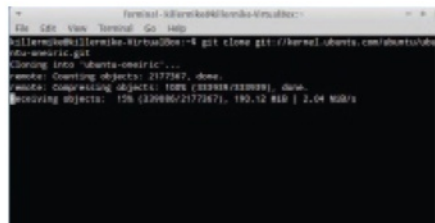
Warning: This is stage where you actually make some changes to your machine. Using our method means that we can install our custom kernel just like any other package. Type 'sudo dpkg -i linux-image[version].deb' and then, 'sudo dpkg -i linux-headers-[version].deb'.



12 Edit GRUB 2

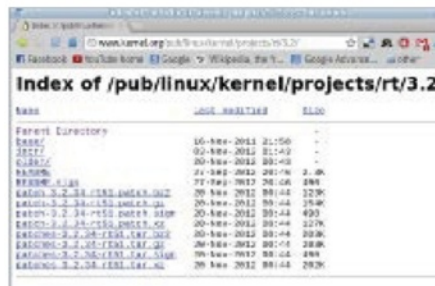
Open up /etc/default/grub in a text editor. Edit the line 'GRUB_HIDDEN_TIMEOUT=0' to remove the zero. Now run 'update-grub'. The Grub Menu will be displayed at startup. The old, stock

kernel can be found under the 'Previous Linux Versions' entry. On reboot, type 'uname -a' to check the kernel version.



13 Fetch the kernel using Git (Ubuntu)

This gives you the latest version of the Ubuntu kernel source. Install Git: 'sudo apt-get install git-core'. Fetch the source with 'git clone git://kernel.ubuntu.com/ubuntu/ubuntu-oneiric.git'. Substitute 'oneiric' with the official distribution name that you are using.



14 Fetch kernel manually

Now that you're familiar with compiling the kernel, consider grabbing a newer kernel from kernel.org, although these kernels don't have the extra additions of Ubuntu. Unpack the kernel with 'tar xjfv linux.<version>.tar.bz2' and repeat the earlier steps.



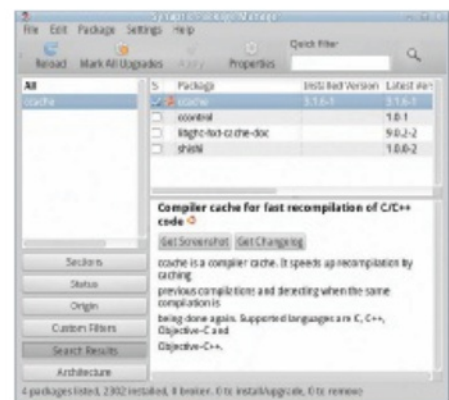
15 Building the kernel on Red Hat

Many of the techniques already applied can be used with Red Hat-derived distributions such as Fedora. Begin by installing the prerequisites: 'su -c 'yum install rpmdevtools yum-utils''. Then type, 'rpmdev-setuptree', and then 'yumdownloader --source kernel' to fetch the kernel. 'su -c 'yum-builddep kernel-[use tab completion].src.rpm'' and then 'rpm -Uvh kernel-[use tab completion].src.rpm'. Move into the SPECS directory with 'cd ~/rpmbuild/SPECS' and type 'rpmbuild -bp --target=\$(uname -m) kernel.spec'. Now cd into the source directory (something like, rpmbuild/BUILD/kernel-3.6.fc17/linux-3.6.8-2.fc17.i686/)



16 Begin the build (Red Hat)

Generate and edit the .config file as before. Open ~/rpmbuild/SPECS/kernel.spec and change the line '#% define buildid' from '.local' to something unique to identify your kernel. From within, type '~/rpmbuild/SPECS/' then 'rpmbuild -bb --without debug debuginfo --target=uname -m kernel.spec' to build.



17 Install ccache

ccache is a compiler cache that is particularly useful when compiling the kernel, although you only start to see the full benefit on repeat compilations. Enter 'sudo apt-get install ccache'. Then type 'export CC="ccache gcc"' and then 'export CXX="ccache g++"'.

18 Cross-compilation (Raspberry Pi)

You can use a powerful machine to 'cross-compile' a kernel for a smaller one. We'll use the Raspberry Pi as an example. You will need a way of accessing files on the Pi storage device. Type 'sudo apt-get install gcc-arm-linux-gnueabi make git-core ncurses-dev' on the PC.

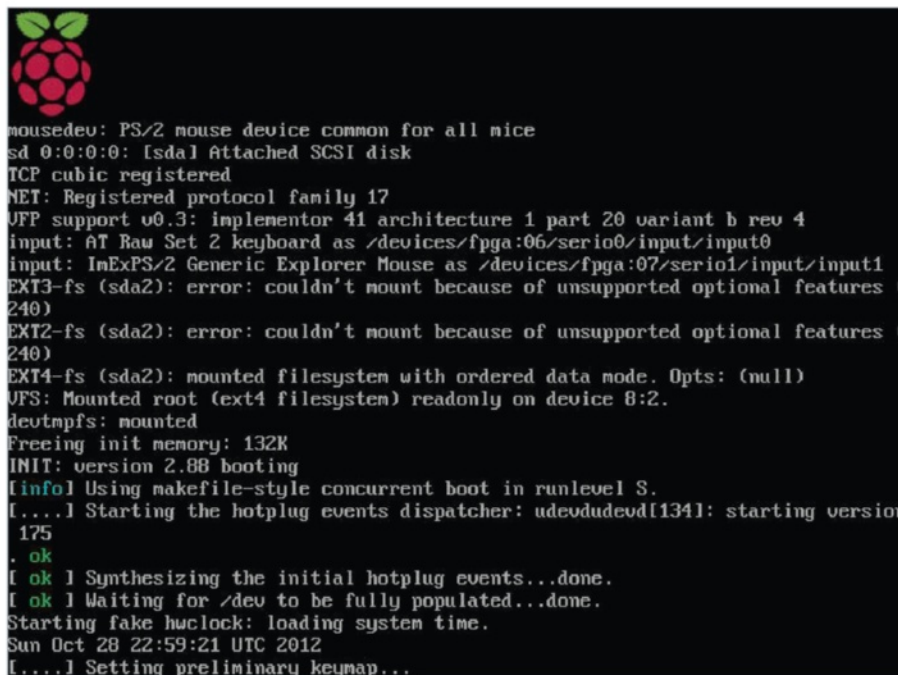


19 Fetch the source

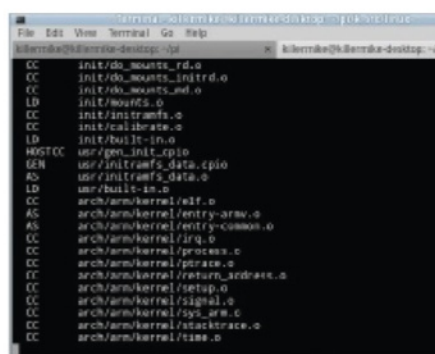
On the PC, create a directory and then move into it. Then type 'git clone https://github.com/raspberrypi/firmware' followed by 'git clone https://github.com/raspberrypi/linux.git'. Now move into the 'linux' directory.

20 Configuration

You can fetch the config from a running Raspberry Pi by typing 'sudo zcat /proc/config.gz > .config' into a shared directory, and then copying it to the 'linux' directory on the PC. Type 'make ARCH=arm CROSS_COMPILE=/usr/bin/arm-linux-gnueabi-oldconfig'.



“The old, stock kernel can be found under the ‘Previous Linux Versions’ entry”



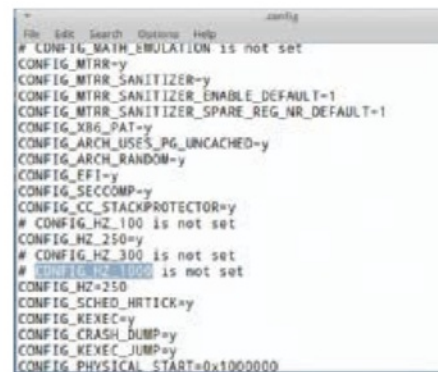
21 Cross-compile kernel

You can then edit the config with 'make ARCH=arm CROSS_COMPILE=/usr/bin/arm-linux-gnueabi-xconfig'. Now run 'make ARCH=arm CROSS_COMPILE=/usr/bin/arm-linux-gnueabi-k' to compile. Place the new kernel (linux/arch/arm/boot/Image /boot/kernel.img) in /boot/kernel.img.



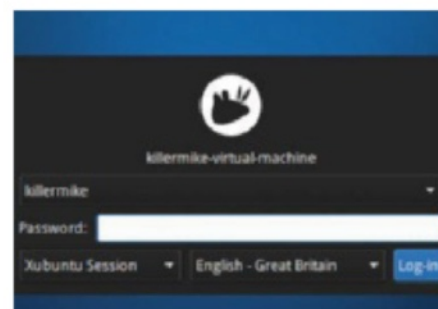
22 Tweak for MIDI performance 1

Here's an example to tweak the kernel for improved MIDI latency for an AV workstation. First, fetch a patch archive at an appropriate level for your kernel from www.kernel.org/pub/linux/kernel/projects/rt/. Repeat the procedure for recreating the source tree, but apply the RT patch before the Ubuntu one.



23 Tweak for MIDI performance 2

Follow the steps as for a normal installation, but first carry out some tweaks in the .config file. Ensure that 'CONFIG_HZ_1000' is set to 'y' and that 'CONFIG_APM' is set to 'n'.



24 Work inside a virtualiser

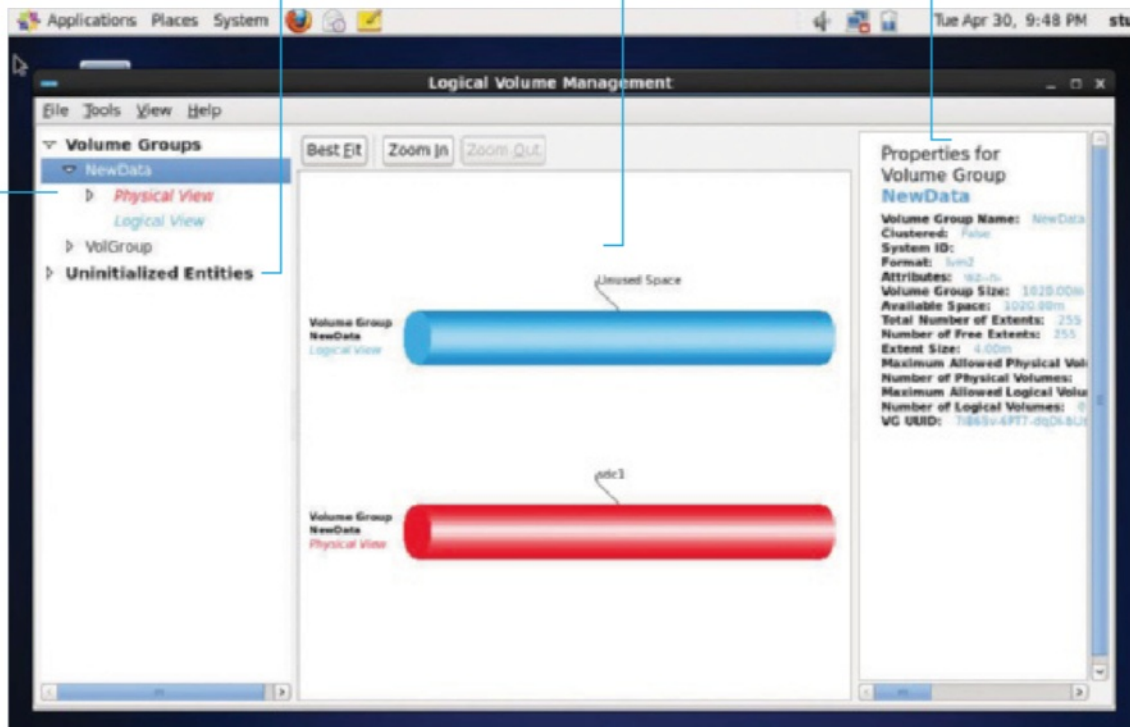
If this is the first time you've built a kernel, you might like to start by working inside a virtualiser (eg QEMU). This gives numerous advantages such as being able to pause the build process. Try to devote as much memory and as many cores as you possibly can.

This gives the make-up of the volume groups and the breakdown of them. It's much more visually appealing than the command-line version

Here you will find all unused members (disks) that can be added into the volume groups and expand the disk space

If you select a volume group, this central visual gives you the make-up of the VG with logical and physical disks

This column gives you advanced information on the make-up of the volume group. It gives some useful information such as size and system attributes



Resize your disks on the fly with LVM

Grow your disks like a true master and never reformat and restore your drive again

Often we see or hear of people running out of disk space on their Linux systems and resorting to resizing tools or worse. Linux has the ability to extend (or shrink) logical partitions across spare space or even across additional disks.

This guide walks you through the why and the how of resizing your disks.

We will be creating an additional new LVM-based partition on our virtual CentOS server, as well as the underlying structures that are called physical volumes (PVs) and volume groups (VGs) that make this awesome technology possible.

These three different parts together make up LVM (Logical Volume Manager). The best way to think about LVM setups is that they are like a layered sponge cake. On the bottom layer we have the physical volumes: the hard disks. Then we have the volume groups, the cake's 'cream' that can in effect cement the disks together and provide a smooth contiguous surface to carve out our partitions. This means that a partition can be bigger than a single disk. Also, when you want to expand a disk, as long as you have space in your volume group, you can.

Resources

VirtualBox: www.virtualbox.org

CentOS 6.4 ISO: wiki.centos.org

```

root@localhost:~# lvs
LV      VG      Attr      LSize  Pool Origin Data%  Move Log Cpy%Sync Convert
lv_root VolGroup -wi-ao--- 13.54g
lv_swap VolGroup -wi-ao--- 1.97g
    
```

01 Get all your disks in a row
Essentially LVM separates the logical volumes from the physical disks with a glue in between called Volume Groups (or VGs for short). Firstly, you may actually be using LVM without realising it! Most newer distros use LVM when they do their install. To see if you are, just open a root console and type `lvs`.

If you see something like that shown in the picture above, then you are already using LVM. In our example we will be creating a new logical volume from scratch, but the same principles around resizing and reducing still apply.

02 Building our new LVM setup
Use the command `fdisk -l` and find the disk we're going to use as the first disk in our LVM setup. In our example, `sdb` is our new disk.

To tell Linux that the disk is going to be an LVM disk, we need to set the disk type (Label) to LVM. Use the command: `fdisk /dev/sdb`

Press 'n' to add a new partition. The type of disk we are adding is a primary, so press 'p' and as we are only putting a single partition on the disk, we can select partition number 1. Accept the geometry the machine suggests by pressing

Enter on the questions. Don't quit out of `fdisk` yet as we have more to do!

```

WARNING: DOS-compatibl mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): n
Command action
  n   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1044, default 1):
Using default value 1
Last cylinder, or (size in K,M,G) (1-1044, default 1044):
Using default value 1044

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e
Changed system type of partition 1 to be Linux LVM

Command (m for help): w
The partition table has been altered:

Calling ioctl() to re-read partition table.
Syncing disks.
[root@localhost ~]# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
[root@localhost ~]#
    
```

03 Preparing the bottom layer of our LVM cake: physical volumes
Once we have set up the partition, we need to identify it as an LVM device. To do this while still in `fdisk`, press 't' to change the disk label, and enter the label ID, which in our case is 8e. Follow

```

File Edit View Search Terminal Help
[stu@localhost Desktop]$ su -
Password:
[root@localhost ~]# lvs
LV      VG      Attr      LSize  Pool Origin Data%  Move Log Cpy%Sync Convert
lv_root VolGroup -wi-ao--- 13.54g
lv_swap VolGroup -wi-ao--- 1.97g
[root@localhost ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xdd31f2d9.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by write)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help):
    
```

“Think of LVM setups being like a layered sponge cake”

this by 'w' (to write the changes to disk). At this point we have created a partition and identified it as an LVM disk type. So now we need to 'bless' or prepare the physical disk upon which LVM will be based. To do this we use a command called `pvcreate`. This effectively prepares and marks the disk as an LVM disk. To do this, use the command: `pvcreate /dev/sdb1`

```

root@localhost:~# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
[root@localhost ~]#
    
```

04 Followed by the LVM jam: volume groups
The bottom 'cake layer' is now created and we can make the jam in the middle: the volume groups. Each of these is like a chunk of storage that you can slice into one or several logical volumes. The difference is that you can expand and reduce disks within the volume group. The `vg_data` bit is the name for the volume group: `vgcreate vg_data /dev/sdb1`

If you want to span multiple disks, just add in the extra devices after `/dev/sdb1`. Remember to mark them as LVM disks using the `pvcreate` command and then marking them as LVM disks in `fdisk`!

```

Evolution
Manage your email, contacts and schedule

File Edit View Search Terminal Help
[root@localhost ~]# vgcreate -L 1G -n lv_data vg_data
Logical volume "lv_data" created
[root@localhost ~]#
    
```

05 Adding the top of the cake: logical volumes
Now that we have created the second layer, we can add the logical volumes: `lvcreate -L 1G -n lv_data vg_data`
This command will create a 1GB logical volume

“Remember, a VG is just a chunk of space”

from the volume group we created. To clarify, lv_data is our new logical volume, while vg_data is the volume group to take it from. You can make the volume as small or large as you like, given a big enough volume group. We are using the entire volume group, rather than just a small portion of it. All the volume groups can be found under the /dev/mapper directory.

Now we can create a logical disk on top of it.

```

root@localhost:~# mkfs.ext4 /dev/mapper/vg_data-lv_data
mkfs2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
62526 inodes, 262144 blocks
13197 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 28 mount
cycles, whichever comes first. Use tune2fs -c or -i to
override.
root@localhost:~#
    
```

06 Making our volumes

Now we can format the disk using the following command:

```
mkfs.ext4 /dev/mapper/vg_data-lvdata
```

To test the partition, try mounting it.

Now this is where things get interesting. This could be a 60GB or 600GB disk. We just chose the 1GB as an example. Now, say we have filled this disk with all our OSs and such, how do we fix it and add that shiny new disk into the logical disk? This is where we see the joy of LVM. We can expand our volume over the new disk and make our volume bigger!

07 Expanding our volumes

To do this, it is a very similar method to creating an LVM; we need to in effect prepare the new disk as an LVM member. To do so, repeat the process we performed earlier to prepare the disks (with fdisk and pvcreate).

```

File Edit View Search Terminal Help
[root@localhost ~]# resize2fs /dev/mapper/vg_data-lv_data
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/mapper/vg_data-lv_data to 524288 (4k) blocks.
The filesystem on /dev/mapper/vg_data-lv_data is now 524288 blocks long.

[root@localhost ~]#
    
```

```

File Edit View Search Terminal Help
[root@localhost ~]# lvextend -L +1G /dev/mapper/vg_data-lv_data
Extending logical volume lv_data to 2.00 GiB
Logical volume lv_data successfully resized
[root@localhost ~]#
    
```

Now that the partition is prepared, we can add it into the volume group we just created. To do this we use the vgextend command, as shown below. The syntax is quite straightforward with the first argument being the volume group you want to add the disk to and secondly, the disk you wish to add.

```
vgextend vg_data /dev/sdc1
```

Now that the VG is extended, we need to perform a similar process to increase the logical disk to use the underlying storage we expanded before. Below is a simple example to just expand our VG with a 1GB partition.

```
lvextend -L +1G /dev/mapper/vg_data /dev/sdc1
```

08 Generation game

And lastly we will need to extend the actual file system. There is a command called resizefs that we can use. As the name implies, it is used to resize the filesystem.

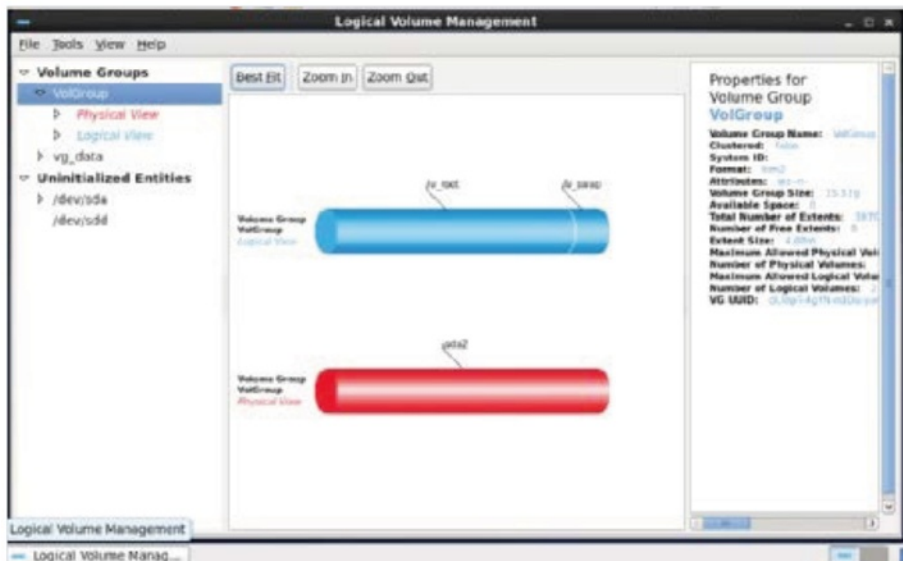
```

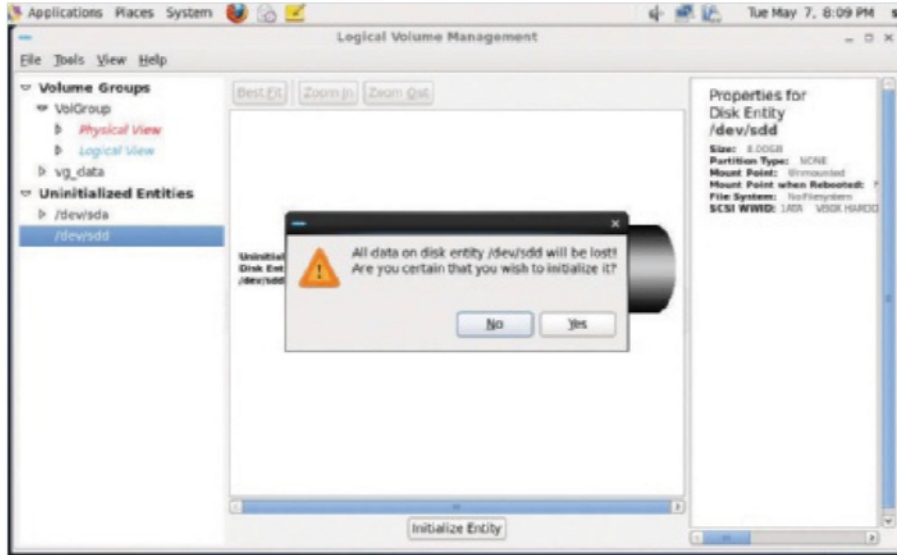
[root@localhost ~]# resize2fs /dev/mapper/vg_data-lvdata
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/mapper/vg_data-lvdata to 2219008 (4k) blocks.
The filesystem on /dev/mapper/vg_data-lvdata is now 2219008 blocks long.
    
```

09 Managing LVMs the GUI way

There is actually a GUI tool, system-config-lvm. We purposely didn't mention this earlier because you need to understand how all the components of how the system fits together. Not only that but also not all systems, especially servers have a GUI installed. To install the package install system-config-lvm.noarch. This will install the graphical LVM tool.

To try it out, let's try with a basic server with an additional 1 GB drive.





10 Creating our physical volumes (again!)

Start by loading the LVM tool. If you look to the left-hand side of the application you will see Volume Groups, Unallocated Volumes, and Uninitialized Entities. It shows the volume groups that we work with, volumes and raw devices that are not members of volume groups.

Doing it this way can be very useful in that it will set up the disk label and such automatically, so no terminal and fdisk are required.

Before you add your unused partition, you will need to initialise the partition, as you can see. Do this by clicking Initialize Entity. This is equivalent to preparing a volume using fdisk and pvcreate as we did before.

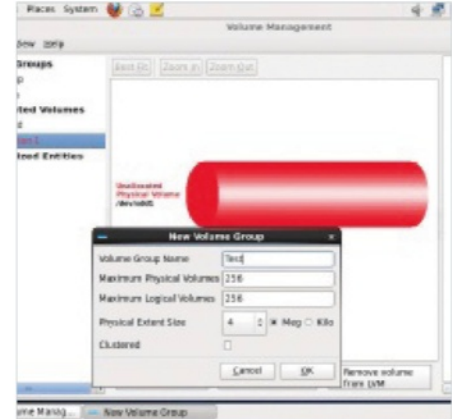
11 ...And the volume group

Once this has completed, you will see that the disk has moved from Uninitialized to Unallocated – so, basically, it is prepped to add to our volume group.

You will also notice that if you click on the unallocated volumes you will get a group of buttons below, each stating its purpose.

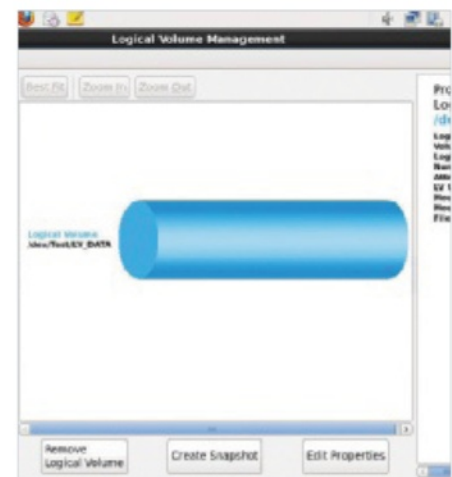
You could, if you wanted, create a new volume group with this disk – or, as we did before, add the disk to an existing volume. Remember, a VG is just a chunk of space that you can split up as you wish; so when you are presented with the sizing screen, carve up that space as you wish.

When creating a new volume group, you will get a selection, as shown below.



12 Creating a new volume group

You can leave these options as they are as they're reasonable defaults. At this point you will see the make-up of your new volume group, the physical disks and the logical make-up.



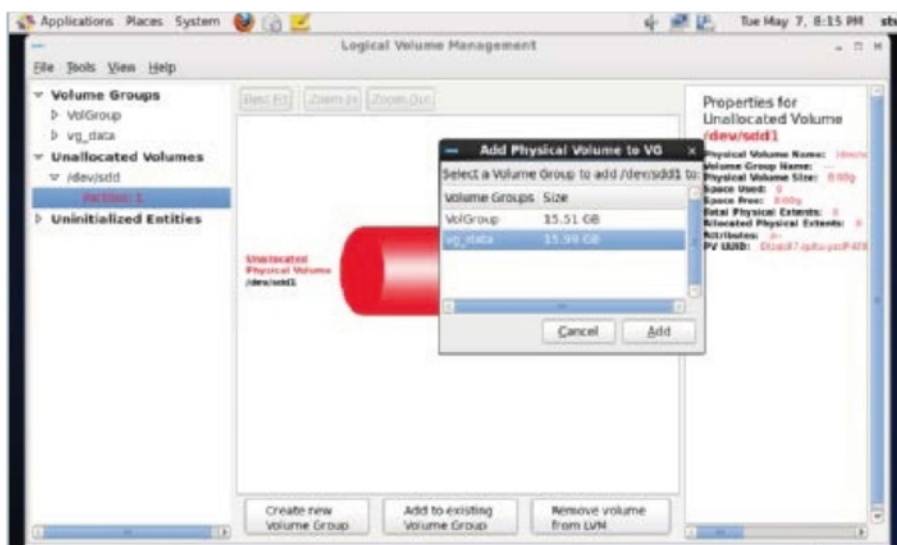
13 Carving out storage from our LVM

The last stage is to carve up a chunk of space, so select the Logical view for the VG you want to carve data out of. Hit the Create Logical Volume button. Now you can specify the logical disks.

So, it's easy enough. Give it a good name. Again for LV properties, just go with the defaults. Size is easy enough and obvious enough. If you want to use the entire disk (most people do), click Use Remaining.

Now we have set the size, we need to decide what file system to use. As we are using CentOS 6, our default file system is ext4. Unless you have a compelling reason to change to it, leave the default selection as it is.

All file systems need a mount point to access them, so let's put in the mount points.



1 Import libraries

These are the libraries we are going to be using for this program

2 Set up variables

These are some variables we'll use to keep track of the script's progress

3 Initialisation

This is the initialising function that we will use to handle the input from the user

Full code listing

```

01 import os, sys, urllib2, argparse, datetime, atexit
    from bs4 import BeautifulSoup

    addresses = []
    deepestAddresses = []

    maxLevel = 1
    storeFolder = "Wikistore " + str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M"))

02 undesirables = [ {"element": "table", "attr": {'class': 'infobox'}}, {"element": "table", "attr": {'class':
    : 'vertical-navbox'}}, {"element": "span", "attr": {'class': 'mw-editsection'}}, {"element": "div", "attr":
    : {'class': 'thumb'}}, {"element": "sup", "attr": {'class': 'reference'}}, {"element": "div", "attr": {'class':
    : 'reflist'}}, {"element": "table", "attr": {'class': 'nowraplinks'}}, {"element": "table", "attr": {'class':
    : 'ambox-Refimprove'}}, {"element": "img", "attr": None}, {"element": "script", "attr": None}, {"element":
    : "table", "attr": {'class': 'mbox-small'}}, {"element": "span", "attr": {"id": "coordinates"}}, {"element":
    : "table", "attr": {"class": "ambox-Orphan"}}, {"element": "div", "attr": {"class": "mainarticle"}}, {"element":
    : None, "attr": {"id": "References"}} ]

    def init():
        parser = argparse.ArgumentParser(description='Handle the starting page and number of levels we're going to
        scrape')
        parser.add_argument('-URL', dest='link', action='store', help='The Wikipedia page from which we will start
        scraping')
        parser.add_argument('-levels', dest="levels", action='store', help='How many levels deep should the scraping
        go')
        args = parser.parse_args()

        if(args.levels != None):
            global maxLevel8
            maxLevel = int(args.levels)

        if(args.link == None):
            print("You need to pass a link with the -URL flag")
            sys.exit(0)
        else:
            if not os.path.exists(storeFolder):
                os.makedirs(storeFolder)

            grabPage(args.link, 0, args.link.split("/wiki/")[1].strip().replace("_", " "))

        atexit.register(cleanUp)

    def isValidLink(link):

```

Scrape Wikipedia with Beautiful Soup

Use the Beautiful Soup Python library to parse Wikipedia's HTML and store it for offline reading

Resources

Beautiful Soup:

www.crummy.com/software/BeautifulSoup/

HTML5Lib:

<https://github.com/html5lib/html5lib-python>

Python 2.6+ & WikiParser.zip

Six: <https://pypi.python.org/pypi/six/>

In this tutorial we'll use the popular Python library Beautiful Soup to scrape Wikipedia for links to articles and then save those pages for offline reading. This is ideal for when travelling or in a location with a poor internet connection.

The plan is simple: using Beautiful Soup with the HTML5Lib Parser, we're going to load a Wikipedia page, remove all of the GUI and unrelated content, search the content for links to other Wikipedia articles and then, after a tiny bit of modification, write them to a file.

Even though it's now the de facto knowledge base of the world, Wikipedia isn't great when it comes to DOM consistency – that is, IDs and classes are sometimes quite loose in their usage. Because of this, we will also cover how to handle all of the excess bits and bobs of the Wikipedia GUI that we don't need, as well as the various erroneous links that won't be of much use to us. You can find the CSS stylings sheet and a Python script pertaining to this tutorial at <http://bit.ly/19MibBv>.

4 Get the page

Here we grab the page we want to store and remove the bits of the document we don't need

5 Check links

Then we iterate through all of the <a> tags and check if there's a valid link to another page we can grab, and tweak them for our own use

6 Copy to file

After that, We take the content we've parsed and put it into a brand new HTML file

7 Clean up

Once every page has been parsed and stored, we'll go on through and try to remove any dead links

8 Initialise

This is how we will initialise our script

```

    if "/wiki/" in link and ":" not in link and "http://"
not in link and "wikibooks" not in link and "#" not in link
and "wikiquote" not in link and "wiktionary" not in link
and "wikiversity" not in link and "wikivoyage" not in link
and "wikisource" not in link and "wikinews" not in link and
"wikiversity" not in link and "wikidata" not in link:
        return True
    else:
        return False

def grabPage(URL, level, name):
    opener = urllib2.build_opener()
    opener.addheaders = [('User-agent', 'Mozilla/5.0')]
    req = opener.open(URL)

    page = req.read()

    req.close()

    soup = BeautifulSoup(page, "html5lib", from_encoding="UTF-8")
    content = soup.find(id="mw-content-text")

    if hasattr(content, 'find_all'):
        global undesirables

        for notWanted in undesirables:
            removal = content.find_
all(notWanted['element'], notWanted['attr'])
            if len(removal) > 0:
                for e1 in removal:
                    e1.extract()

            also = content.find(id="See_also")

            if(also != None):
                also.extract()
                tail = also.find_all_next()
                if(len(tail) > 0):
                    for element in tail:
                        element.extract()

            for link in content.find_all('a'):

                href = link["href"]

                if isValidLink(href):

                    if level < maxLevel:

                        stored = False;
                        for addr in

addresses:
                            if addr
== link.get("href"):
                                stored = True

                                if(stored == False):
                                    title =
link.get('href').replace("/wiki/", "")
                                addresses.append(str(title + ".html"))

                                grabPage("http://en.wikipedia.org" + link.get('href'), level + 1,
                                title)
                                    print title

                                link["href"] = link["href"].replace("/
wiki/", "") + ".html"

```

04

05

Full code listing continued

```

        fileName = str(name)

        if level == maxLevel:
            deepestAddresses.append(fileName.
replace('/', '_') + ".html")

            doctype = "<!DOCTYPE html>"

            head = "<head><meta charset='UTF-8' /><title>" +
fileName + "</title></head>"

            f = open(storeFolder + "/" + fileName.replace('/',
'_') + ".html", 'w')
            f.write(doctype + "<html lang='en'>" + head +
"<body><h1>" + fileName + "</h1>" + str(content) + "</body></
html>")
            f.close()

def cleanup():
    print("\nRemoving links to pages that have not been
saved\n")

    for deepPage in deepestAddresses:

        rF = open(storeFolder + "/" + deepPage, 'r')

        deepSoup = BeautifulSoup(rF.read(), "html5lib",
from_encoding="UTF-8")

        for deepLinks in deepSoup.find_all('a'):
            link = deepLinks.get("href")

            pageStored = False

            for addr in addresses:
                if addr == link:
                    pageStored = True

            if pageStored == False:

                if link is not None:

                    if '#' not in
link:
                        del
deepLinks['href']
                    elif '#' in link
and len(link.split('#')) > 1 or ':' in link:
                        del
deepLinks['href']

                    wF = open(storeFolder + "/" + deepPage, 'w')
                    wF.write(str(deepSoup))
                    wF.close()

            print("Complete")

if __name__ == "__main__":
    init()

```

06

07

08

“Wikipedia isn't great when it comes to DOM consistency”

02

```

import os, sys, urllib2, argparse, datetime, atexit
from bs4 import BeautifulSoup

addresses = []
deepestAddresses = []

maxLevel = 1
storeFolder = "Wikistore" + str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M"))

undesirables = [ ("element": "table", "attr": {"class": "infobox"}), ("element": "table", "attr": {"class": "vertical-navbox"}), ("element": "span", "attr": {"class": "mw-editsection"}), ("element": "div", "attr": {"class": "thumb"}), ("element": "sup", "attr": {"class": "reference"}), ("element": "div", "attr": {"class": "reflist"}), ("element": "table", "attr": {"class": "nowiki"}), ("element": "table", "attr": {"class": "mbox-Refimprove"}), ("element": "img", "attr": None), ("element": "script", "attr": None), ("element": "table", "attr": {"class": "mbox-small"}), ("element": "span", "attr": {"id": "coordinates"}), ("element": "table", "attr": {"class": "mbox-Orphan"}), ("element": "div", "attr": {"class": "mainarticle"}), ("element": None, "attr": {"id": "References"}) ]

def init():
    parser = argparse.ArgumentParser(description='Handle the starting page and number of levels we're going to scrape')
    parser.add_argument('-URL', dest='link', action='store', help='The Wikipedia page from which we will start scraping')
    parser.add_argument('-levels', dest='levels', action='store', help='How many levels deep should the scraping go')
    args = parser.parse_args()

    if(args.levels != None):
        global maxLevel
        maxLevel = int(args.levels)

    if(args.link == None):
        print("You need to pass a link with the -URL flag")
        sys.exit(0)
    else:
        if not os.path.exists(storeFolder):
            os.makedirs(storeFolder)

        grabPage(args.link, 0, args.link.split("/wiki/")[1].strip().replace(".", ""))

    atexit.register(cleanUp)

def isValidLink(link):
    if "/wiki/" in link and "." not in link and "http://" not in link and "wikibooks" not in link and "W" not in link and "wikisource" not in link and "wikinews" not in link and "wikiversity" not in link and "wikivoyage" not in link and "wikisource" not in link and "wikinews" not in link and "wikiversity" not in link and "wikidata" not in link:
        return True
    else:
        return False
                
```

WIKI-EVERYTHING

Wikipedia has so many different services that interlink with each other; however, we don't want to grab those pages, so we've got quite a lengthy conditional statement to stop that. It's pretty good at making sure we only get links from Wikipedia.

03

INFINITE LINKS

Wikipedia has a lot of links and when you start following links to links to links, the number of pages you have to parse can grow exponentially, depending on the subject matter. By passing through the levels value, we put a cap on the amount of pages we can grab— although the number of files stored can still vary greatly. Use it wisely.

01 Install Beautiful Soup & HTML5Lib

Before we can start writing code, we need to install the libraries we'll be using for the program (Beautiful Soup, HTML5Lib, Six). The installation process is fairly standard: grab the libraries from their respective links, then unzip them. In the terminal, enter the unzipped directory and run `python setup.py install` for each library. They will now be ready for use.

02 Creating some useful variables

These variables will keep track of the links we've accessed while the script has been running: `addresses` is a list containing every link we've accessed; `deepestAddresses` are the links of the pages that were the furthest down the link tree from our starting point; `storeFolder` is where we will save the HTML files we create and `maxLevel` is the maximum depth that we can follow the links to from our starting page.

03 Handling the user's input

In the first few lines of this function, we're just creating a helper statement. Afterwards, we're parsing any arguments passed into the program on its execution and looking for a `-URL` flag and a `-levels` flag. The `-levels` flag is optional as we already have a preset depth that we'll follow the links to, but we need a link to start from so if the `-URL` flag is missing, we'll prompt the user and exit. If we have a link, then we quickly check whether or not we have a directory to store files in – which we'll create if we don't – and then we'll fire off the function to get that page. Finally, we register a handler for when the script tries to exit. We'll get to that bit later.

04 Retrieving the page from the URL

Here we're using `urllib2` to request the page the the user has asked for and then,

once we've received that page, we're going to pass the content through to Beautiful Soup with the `soup` variable. This gives us access to the methods we're going to call as we parse the document.

05 Trimming the fat

Wikipedia has a lot of nodes that we don't want to parse. The content variable allows us to straight away ignore most of Wikipedia's GUI, but there are still lots of elements that we don't want to parse. We remedy this by iterating through the list 'undesirables' that we created

“Wikipedia has a lot of nodes that we don't want to parse”

06

```

for link in content.find_all('a'):
    href = link["href"]
    if isValidLink(href):
        if level < maxLevel:
            stored = False;
            for addr in addresses:
                if addr == link.get("href"):
                    stored = True

            if(stored == False):
                title = link.get('href').replace("/wiki/", "")
                addresses.append(str(title + ".html"))
                grabPage("http://en.wikipedia.org" + link.get('href'), level + 1, title)
                print title

        link["href"] = link["href"].replace("/wiki/", "") + ".html"

fileName = str(name)

if level == maxLevel:
    deepestAddresses.append(fileName.replace('/', '_') + ".html")

doctype = "<!DOCTYPE html>"

head = "<head><meta charset='UTF-8' /><title>" + fileName + "</title></head>"

f = open(storeFolder + "/" + fileName.replace('/', '_') + ".html", 'w')
f.write(doctype + "<html lang='en'>" + head + "<body><h1>" + fileName + "</h1>" + str(content) + "</body></html>")
f.close()

```

STYLING

Currently, the HTML page will use the built-in browser styles when rendering the page. If you like, you can include the style sheet included in the tutorial resources to make it look a little nicer. To use it, you can minify the script and include it inside a `<style>` tag in the head string on line 102, or you can rewrite the head string to something like:

```

head = "<head><meta charset='UTF-8' /><title>"
+ fileName + "</title><style>"
+ str(open("/PATH/TO/STYLES", 'r').read()) + "</style></head>"

```

earlier on in the document. For each different div/section/node that we don't want, we call BeautifulSoup's `find_all()` method and use the `extract()` method to remove that node from the document. At the end of the undesirables loop, most of the content we don't want any more will be gone. We also look for the 'also' element in the Wiki page. Generally, everything after this div is of no use to us. By calling the `find_all_next()` method on the also node, we can get a list of every other element we can remove from that point on.

06 Grabbing the links

By calling `content.find_all('a')` we get a list of every `<a>` in the document. We can iterate through this and check whether or not there is a valid Wikipedia link in the `<a>`'s href. If the link is a valid link, we quickly check how far down the link tree we are from the original page. If we've reached the maximum depth we can go, we'll store this page and call it quits, otherwise we'll start looking for links that we can grab within it. For every page we request, we append its URL

“Beautiful Soup is a fast, elegant framework that works with a number of Python HTML parsers”

to the addresses list; to make sure we don't call the same page twice for each link we find, we check if we've already stored it. If we have, then we'll skip over the rest of the loop, but if we've not then we'll add it to the list of URLs that we've requested and fire off a request. Once that check is done, We then do a quick string replace on that link so that it points to the local directory, not to the subfolder `/wiki/` that it's looking for.

07 Writing to file

Now we create a file to store the newly parsed document in for later reading. We change any `/` in the filename to `_` so the script doesn't try and write to a random folder. We also do a quick check to see how many links we've followed since the first page. If it's the max level, we'll add it to the `deepestAddresses` list. We'll use this a little bit later.

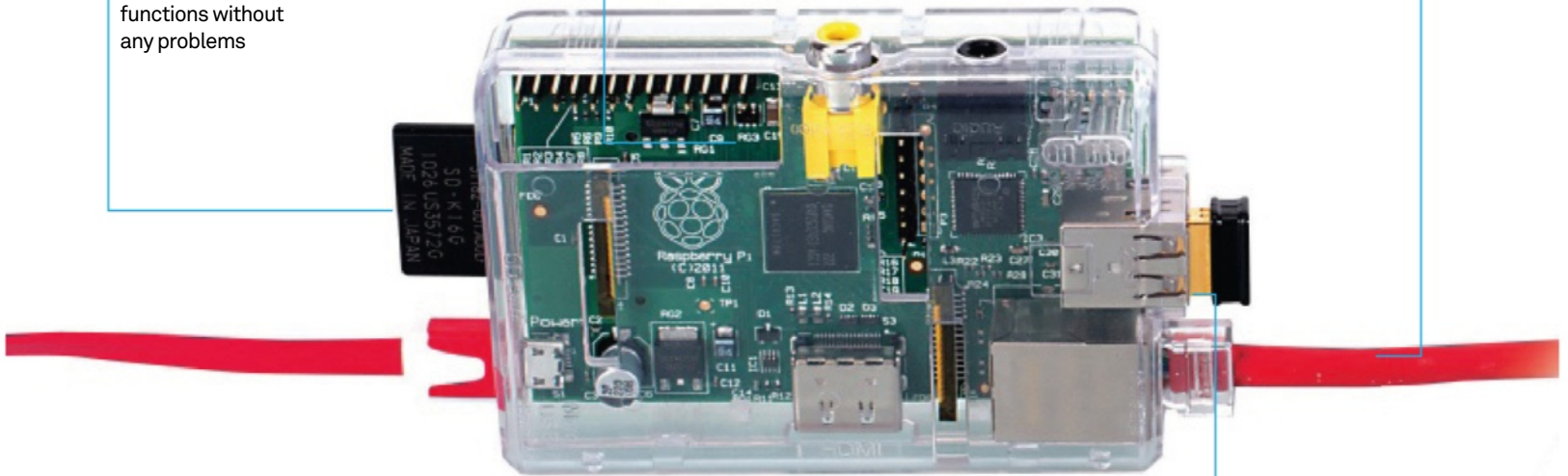
08 Tying up loose ends

After our script has iterated through every link on every page to the maximum level of depth that it can, it will try to exit. On line 34 of the code (on the disc and online) in the `init` function, we registered the function `cleanUp` to execute on the program trying to exit; `cleanUp`'s job is to go through the documents that we've downloaded and check that every link we've left in the pages does in fact link to a file that we have available. If it can't match the link in the href to a file in the addresses list, it will remove it. Once we're done, we will have a fully portable chunk of Wikipedia we can take with us.

This isn't a hardware hack – a spare SD card can be used for the Tor router, and other SD cards can be used for different functions without any problems

Using not much more than a Raspberry Pi, you can route one or more systems through a Tor-enabled access point, guaranteeing anonymity

Hook into the internet just about anywhere there's an internet connection – a relative's house, hotel rooms and more



Turn your Raspberry Pi into a secure router

Surf the internet privately with Onion Pi and keep your system safe and your browsing anonymous, wherever you are

Connect everything over a wireless network – no need to directly connect to the Pi with a cable

Resources

A Raspberry Pi

Raspbian: www.raspberrypi.org/downloads

Compatible Wi-Fi adaptor:

www.adafruit.com/products/814

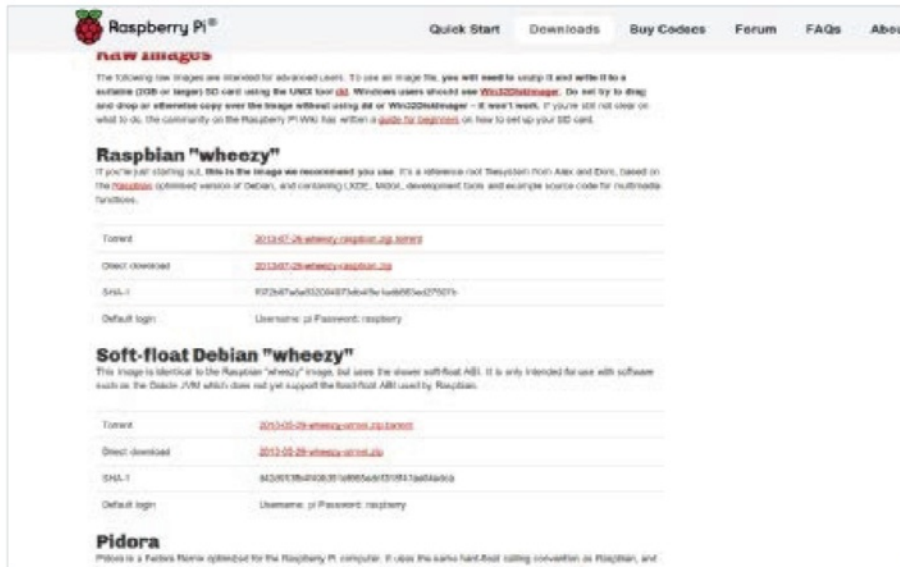
It's possible to turn your Raspberry Pi into the ultimate portable wireless router, as it requires very little power and gives you a wireless network wherever there's the most basic of internet connections. What if it's not enough to know you can search the web, though? What if you want to be wholly secure as you do it? Then it's time to upgrade the router with Tor to protect your privacy on the internet.

This 'Onion Pi', as dubbed by Adafruit, combines Raspbian and Tor to create and secure a wireless access point using just a Raspberry Pi. This project is fairly straightforward: after setting up the wireless access point, we install Tor

and do some basic setup tasks so that it routes traffic properly, and securely. This will keep you anonymous online – a handy feature in a time of privacy concerns all around the web.

When the Pi is not connected to the internet, it should still function as a wireless router, allowing at the very least a wireless LAN in your location.

“It's time to upgrade your router with Tor”



01 Install Raspbian

Raspbian is the Raspberry Pi distro we'll be using for the Onion Pi. Download the zip file, extract the image and then apply it to an SD card using:

```
$ dd bs=4M if=[version number]-wheezy-raspbian.img of=/dev/[SD card location]
```

You can also use NOOBS to install Raspbian if you wish.

02 Set up Raspbian

Go through the initial Raspbian setup and make sure to turn on the SSH server, and to disable autoboot to desktop – this is unnecessary and will only use extra power. You can also tell it to fill up the rest of the card if there's room for it.

03 Pi IP

We'll be accessing your Raspberry Pi via SSH to set it up. To do this we need to know its IP address – you can find it by typing `ifconfig` into the command line. Make a note of it and turn off your Pi.



04 SSH connection

Plug your USB wireless adapter into the Pi and turn it back on. On another computer connected to the same network, open a terminal or type into the command line:

```
$ ssh [user]@[IP address]
```

Then enter the password for your Raspbian if it asks for it.



05 Install DHCP

To make life easier for any system connecting to the Pi access point, we need to install a DHCP server to it. We do this with:

```
$ sudo apt-get install hostapd isc-dhcp-server
```

DHCP will automatically assign IP addresses to network-attached devices, meaning you won't need static IPs.



06 Set up DHCP

Now we need to configure the DHCP server. Edit the configuration file with:

```
$ sudo nano /etc/dhcp/dhcpd.conf
```

And start by putting a `#` in front of the two `option domain-name` entries, then remove the `#` in front of `authoritative`, seven lines down.

07 Server address

At the end of the configuration file, add the following:

```
subnet 192.168.42.0 netmask 255.255.255.0 {
    range 192.168.42.10 192.168.42.50;
    option broadcast-address 192.168.42.255;
    option routers 192.168.42.1;
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "local";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Save and exit.



08 DHCP server

Edit the server configuration files so that it's set to work in conjunction with the wireless adaptor:

```
$ sudo nano /etc/default/isc-dhcp-server
```

Scroll to `INTERFACES` and change it to:

```
INTERFACES="wlan0"
```

09 Incoming Wi-Fi

We need to set up the Wi-Fi adaptor to be both static and accept incoming signals. First:

```
$ sudo nano /etc/network/interfaces
```

Put a # in front of `iface wlan0` and following lines with `wpa roam`, `iface default` and any other affecting `wlan0`.

```
pi@raspberrypi:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/netw
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
```

10 Static IP

Now give the wireless interface a static IP – after the line `allow-hotplug wlan0`, enter the following:

```
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

Save and exit, and then set `wlan0`'s address with:

```
$ sudo ifconfig wlan0 192.168.42.1
```

```
pi@raspberrypi:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/netw
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
up iptables-restore < /etc/iptables.ipv4.nat
```

11 WLAN creation

We need to create a new file that holds all the information for our wireless network. We are going to make it password protected so that only the people we want to can access it. To create the file, start with:

```
$ sudo nano /etc/hostapd/hostapd.conf
```

And then enter the text from the next step.

```
pi@raspberrypi:~
File Edit View Search Terminal Help
GNU nano 2.2.6
interface=wlan0
driver=rtl871xdrv
ssid=LUDPi
hw_mode=g
channel=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=lettherightonein
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

12 WLAN configuration

```
interface=wlan0
driver=rtl871xdrv
ssid=[access point name]
hw_mode=g
channel=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=[password]
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

13 Hostapd

After saving and exiting, we need to edit `hostapd` to point it to this new file. Open it with:

```
$ sudo nano /etc/default/hostapd
```

And then find the line `#DAEMON_CONF=""`. Remove the #, and change it to:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

```
pi@raspberrypi:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/default/hostapd
# Defaults for hostapd init script.
# See http://www.debian.org/doc/debian-policy/section-variables.en.html
# Document and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot, or manual configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
DAEMON_CONF="/etc/hostapd/hostapd.conf"
# Additional daemon options to be appended to hostapd command:
# -d show more debug messages, 1-44 for user level
# -E include key data in debug messages
# -t include timestamps in some debug messages
# Note that -e (daemon mode) and -p (pidfile) options are automatically
# configured by the init.d script, and must not be added to DAEMON_OPTS.
DAEMON_OPTS=""
```

14 Network addressing

Setting up a NAT will allow multiple clients to connect. To do this, run:

```
$ sudo nano /etc/sysctl.conf
```

And add to the bottom of the file:

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

Save this, and then finish by running:

```
$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

```
pi@raspberrypi:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/sysctl.conf
# Do not accept IP source route packets (we are not a router)
net.ipv4.conf.all.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
# Log Martian packets
net.ipv4.conf.all.log_martians = 1
# rpl tweaks
vm.swappiness=1
vm.vfs_free_kbytes = 8192
net.ipv4.ip_forward=1
```

15 IP tables

Run the following three commands to make sure the internet connection is forwarded correctly:

```
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
```

16 Apply configuration

So that this still works after a reboot, type:

```
$ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Then add to the end of `/etc/network/interfaces`:

```
up iptables-restore < /etc/iptables.ipv4.nat
```

```
pi@raspberrypi:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
up iptables-restore < /etc/iptables.ipv4.nat
```

17 Wi-Fi final

Finally, set it up as a daemon so it runs at boot with the following commands:

```
sudo service hostapd start
sudo service isc-dhcp-server start
sudo update-rc.d hostapd enable
sudo update-rc.d isc-dhcp-server enable
```

And the wireless access point part will be finished.

18 Install Tor

After a reboot, we now need to install Tor. Do this simply with:

```
$ sudo apt-get install tor
```

Once it's installed, you'll need to edit the Tor config file with:

```
$ sudo nano /etc/tor/torrc
```

Follow the next step to add all the necessary information to it.

19 Tor configure

Put this below the FAQ comment:

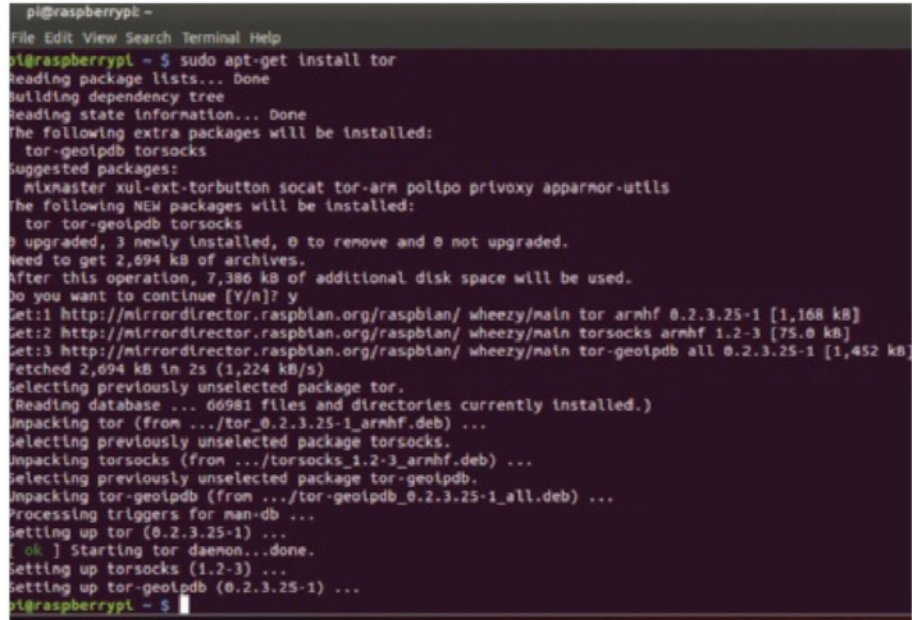
```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.42.1
DNSPort 53
DNSListenAddress 192.168.42.1
```



20 Table flush

We now need to flush the current IP tables so that we can get the routing to go through Tor. First of all, do:

```
$ sudo iptables -F
$ sudo iptables -t nat -F
```



If you want to keep SSH open to connect remotely, you'll need to make an exception for that with:

```
$ sudo iptables -t nat -A PREROUTING
-i wlan0 -p tcp --dport 22 -j
REDIRECT --to-ports 22
```

21 Reroute

Route all DNS traffic first, using:

```
$ sudo iptables -t nat -A PREROUTING
-i wlan0 -p udp --dport 53 -j
REDIRECT --to-ports 53
```

And then route any TCP traffic with:

```
$ sudo iptables -t nat -A PREROUTING
-i wlan0 -p tcp --syn -j REDIRECT
--to-ports 9040
```



22 Check and save

You can check the table setup with:

```
$ sudo iptables -t nat -L
```

If you're happy with it, save it to the NAT file like

before with:

```
$ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

23 Logging

We should create a log file in case you need to debug later. To do this, use these three commands:

```
$ sudo touch /var/log/tor/notices.log
$ sudo chown debian-tor /var/log/tor/notices.log
$ sudo chmod 644 /var/log/tor/notices.log
```

You can also check it with:

```
$ ls -l /var/log/tor
```

24 Secure the router

Finally, we can activate the Tor service so that we can start using the access point securely with:

```
$ sudo service tor start
```

You can check this if you wish with:

```
$ sudo service tor status
```

To make it turn on at boot, you simply add it to rc.d with:

```
$ sudo update-rc.d tor enable
```

Apps

Great distros and applications

168 **openSUSE 13.1 RC 1**
The next step in the Linux distribution for everyone to use

170 **Linux Mint 16 RC**
The Cinnamon-flavoured revolution is here

172 **Fedora 19 Schrödinger's Cat**
How has the latest edition of Fedora shaped up?

174 **Mageia 3 RC**
Pretty much the perfect version of the already great distro

176 **AV Linux 6.0.1**
A highly customised Debian designed for video and audio professionals

178 **Bodhi Linux**
An incredibly lightweight and highly customisable distro

180 **Geany**
A fully featured IDE that's a little more lightweight than most

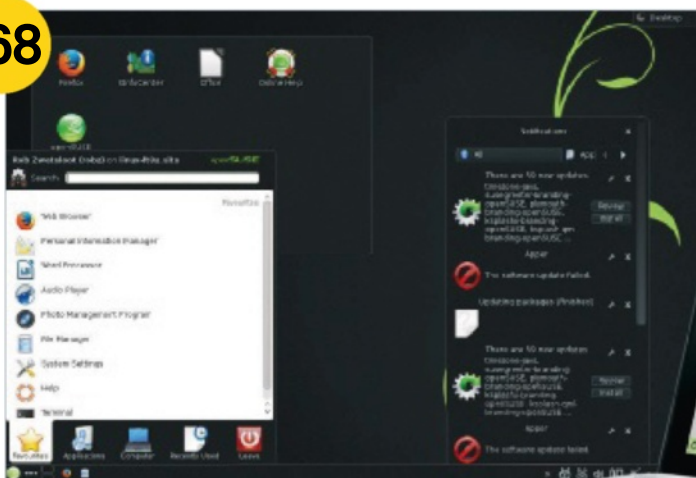
181 **Eclipse**
How does it fare against more community-run efforts?

182 **Parted Magic**
A recovery distro with a focus on hard drive partitioning

183 **SystemRescueCD**
Great for recovery and general maintenance

184 **Dropbox**
No introductions necessary for the king of cloud storage

168



170





185

185 SpiderOak
A veteran in Linux cloud storage

186 OpenShot
An intuitive yet professional movie-making option

187 Kdenlive
A full-featured video editor – it’s the complete package

188 Clementine
A Linux favourite, how is the latest Clementine player?

189 Banshee
Not as popular as Clementine, but still a great option

190 gThumb
A feature-packed photo manager complete with editing options

191 digiKam
A fantastic and comprehensive photo managing application

“Some of the best free and open-source applications around”



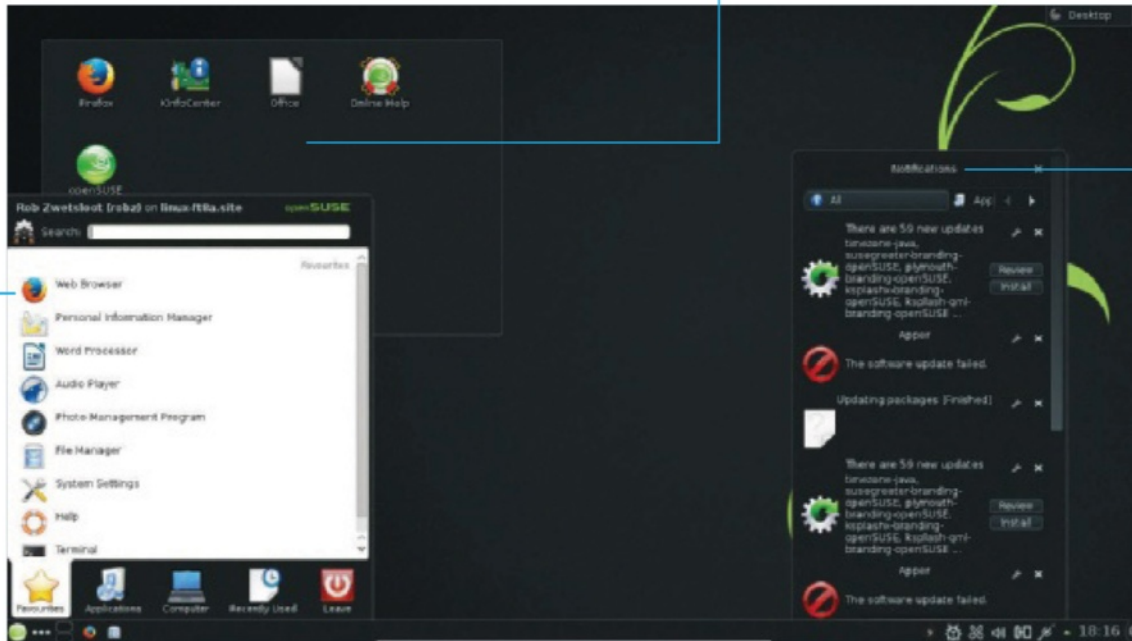
189



App selection in openSUSE is very good, with a packed selection of default apps and a full repository

KDE and GNOME are still the standard desktop environments, and have been themed and tweaked by the developers

There are a selection of standard updates and bug fixes, but YaST is now completely written in Ruby



openSUSE 13.1 RC 1

A look at the release candidate for openSUSE's 13, the next step in the Linux distribution for everyone to use

Pros

YaST is now built in Ruby, allowing for easier development, and is still as easy to use as ever

Cons

No live booting off the full DVD, and a limited selection of official desktops compared to some distros

The 13.x line of openSUSE releases is just about here, ready to move beyond the troubled development woes that the community experienced last year for the early releases of 12.x. While there are only a few changes coming to the next version of openSUSE over the previous ones, there are some wide-reaching effects to various levels of users.

One of the most important changes implemented in 13.1 is porting YaST to Ruby. Previously, the openSUSE control centre software was built in its own proprietary language, meaning few people in the community were able to easily contribute to its code. The port to Ruby has been a straight job, and it was introduced as part of

the distro during a beta version of 13.1. For the desktop user, this may not mean much, but to the developers and the community it's a huge step forward in allowing one of the major features of openSUSE to be much more open and friendly to those who want to commit changes.

With the first iteration of such a port let loose in the wild, it's natural to be concerned over the new YaST's stability and quality – thankfully, the porters seem to have done an exceedingly good job. The control centre is as usable as it's ever been, and there were no issues using it for adding and removing software, changing network settings, adjusting the boot menu and all the other tasks it can perform.



■ Installation from the full DVD allows for a complete and customisable install, while the live CDs offer a great preview

“OpenSUSE is also about community, and the changes to YaST and efforts made with Btrfs are a great indicator of how strong it currently is”

Minor updates

As well as YaST, there’s been some great improvements over Btrfs, the future file system that keeps being just out of reach. It’s not a default yet, but the developers and community have been making an effort to improve its support in the latest version of openSUSE with some impressive results. Right now it’s considered safe to use, with the intention that it’ll be a default in 13.1. We’ve heard that line before, though, about Btrfs from other developers.

There’s a host of updates to all the desktop environments, the Linux kernel has been updated to version 3.11.3 (with the added Btrfs patches) and interestingly, there’s an effort to update GStreamer from 0.1.0 to the newer 1.0 – although this hasn’t been implemented as of the release candidate.

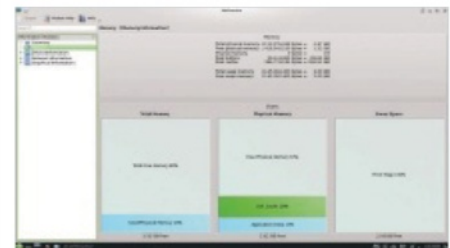
Aside from the big changes to YaST, it’s somewhat of a safe update for openSUSE. There’s no problem with this, of course, though, and it’s allowed it to stay rock solid and compatible with

a lot of hardware types and keep its great user experience intact.

Rousing performance

Updates and new features aside, openSUSE 13.1 still works as advertised. The images supplied come in three main flavours – two live discs containing one of the two main desktop environments, and the full installation DVD. The KDE and GNOME spins allow you to live-boot into openSUSE and give it a test before committing to installing, while the DVD version is specifically just for installation.

The DVD installer is still one of the better Linux installers out there. The dedicated process is split up into distinct sections with a logical flow to the process. Default options are passable for the lower-end users, while there’s plenty of room for customisation and further setup for the more advanced users that encompass the targeted user base of openSUSE itself. You can also choose between the main supported desktops, or select a



■ System information is easily accessible, allowing for system diagnostics on every level

more lightweight alternative if you require it. The only thing really missing is adding or removing different software packs, the kind that the Mageia installer provides. While you can make your own custom ISO that will do this for you with SUSE Studio, it would be nice to have even a basic version of it with the official release.

Installation is quite fast, and will automatically restart and dump you into the desktop. The openSUSE desktop themes continue to be some of the best around, with great aesthetics and design ethos that eke a little bit more out of the standard KDE and GNOME.

The next generation

Right now, then, everything looks fantastic for the next openSUSE. The philosophy of the distro has always been about making it the best OS to use for novices and veterans alike. This is again accomplished with a fantastic selection of tools for sysadmins to manage the systems locally or remotely, and a smart design that allows normal desktop users to quickly get into a new workflow rhythm. OpenSUSE is also about community, and the changes to YaST and efforts made with Btrfs are a great indicator of how strong it currently is. We look forward to seeing what the next version brings.

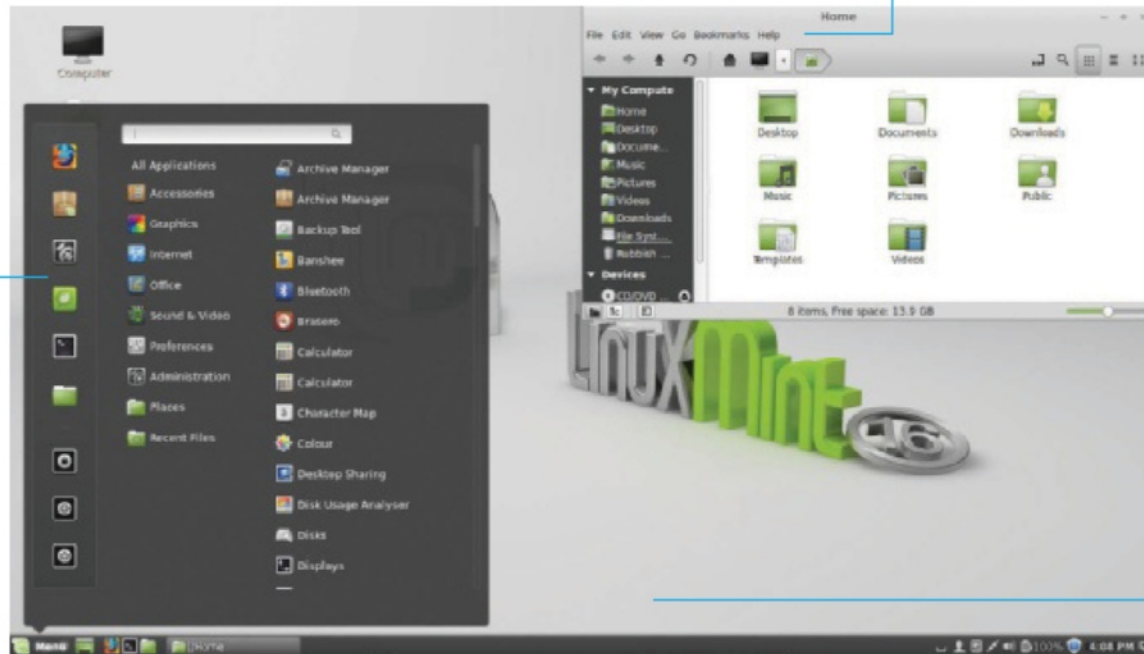


The community distro returns with a new version of YaST and a series of updates that still allow it to be one of the most usable and stable distros available. The 13.x line of openSUSE may be the best yet.

Slight changes to the transparency effects and colours make navigating the already great Mint Menu just that bit better

Edge tiling is improved with a new lock feature that means windows you wish to stay visible always will

Cinnamon 2.0 is very stable, even with its own brand new back-end



Linux Mint 16 RC

The Cinnamon-flavoured revolution is here as Mint releases its first truly independent desktop environment

Pros

Cinnamon is ever so slightly yet noticeably better, even though it's had more work on the back-end than front-end

Cons

Mainly a stepping stone to Linux Mint 17, and you still need a more modern system to make the most of Cinnamon

From a normal user-perspective, Linux Mint 16 might be the lightest Mint release in terms of new features and content. There are some aesthetic changes to Cinnamon, MATE is still roughly the same and there's a new default theme for MDM. It's still the same old, fantastic Linux Mint as before in that regard; however, this seemingly minor release has one of the biggest updates in Linux Mint history. Cinnamon 2.0, the Mint-developed desktop environment, has replaced its GNOME base with one of its own.

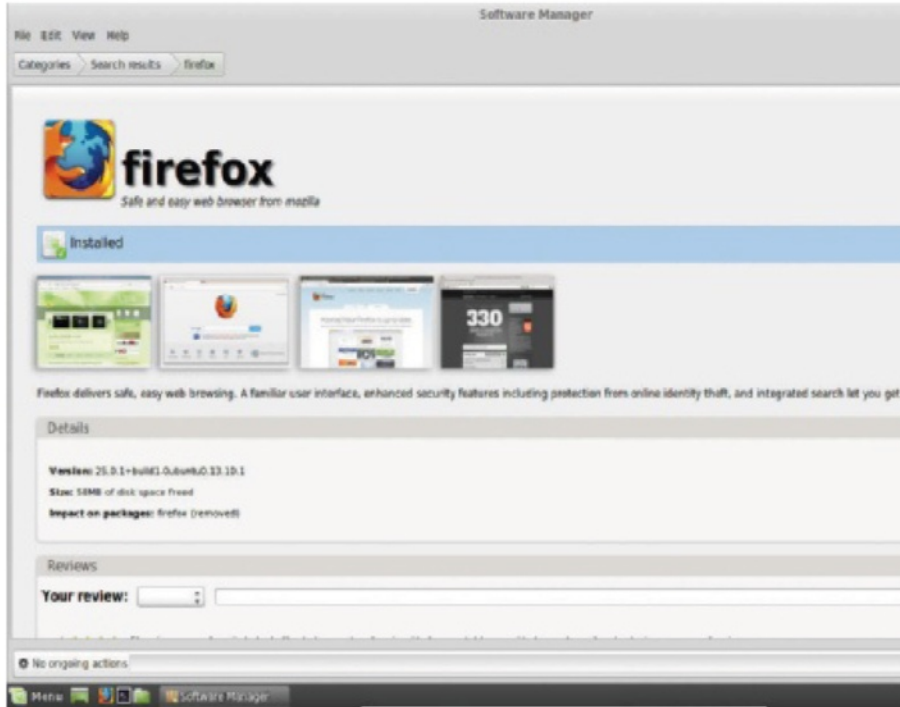
We'll get to that, though – first of all, here are the facts. Linux Mint 16 is based on Ubuntu 13.10, the pre-LTS release, and as usual gets rid of some of the more commercial stuff while keeping the non-free software, making it a better experience for users not too concerned with strictly using FOSS. Cinnamon and MATE versions are offered to start off, with specific Mint-themed KDE and Xfce versions to come.

These desktops are already available through the software repositories, though.

The installation hasn't changed at all since last time, using the same standard Ubuntu installer. It's fast, fairly easy to use and has some decent default options for installing alongside or replacing a distro altogether. It's themed to Mint at least, so there's no mistaking what you're installing.

Cinnamon Spiced

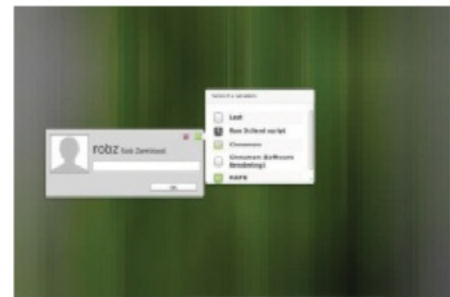
Once installation is completed, though, you restart into one of the first new changes – the new HTML 5 login theme. It's a nice little touch, and it looks little like the Ubuntu login screen. This new MDM theme also has your user selected by default, rather than the previous method of having to manually enter a username or select a user. The previous clouds theme is still available, nonetheless, and you can also switch to a more



■ The Software Manager is one of the things that separates it from Ubuntu, with no ads or paid apps. It's had a few minor updates, including the ability to show off more screenshots



■ MATE is still at 1.6, but it's a great desktop and a solid choice for a Mint install nonetheless



■ MDM has come on in leaps and bounds since its first introduction, and this release is the first time it's seemed truly modern

“We couldn't have asked much more of Linux Mint for this release”

secure login screen if the idea of having your username on display worries you.

Here's where the most interesting part starts – Cinnamon 2.0. On first impressions, long-term Cinnamon users will notice that the icons and Mint Menu are just ever so slightly different: brighter colours and better transparency effects, although the fonts look a touch fuzzier. Generally though, it just highlights the different important areas a bit better, slightly aiding in navigation and workflow, especially for new users. While these are only little touches, and there are a more of them scattered throughout, the biggest change is the aforementioned change to a pure Cinnamon base.

While invisible to most, the removal of the GNOME back-end from the Cinnamon code is an enormous achievement for the Mint team, especially as the final result is very stable. No functionality has been lost in the transition and while nothing has been specifically added either, it means that future versions of Cinnamon will be able to include more innovation than before.

Such as new feature edge-snapping, allowing you to lock a window to a specific corner or side of the desktop and other windows will maximise around it. This is great for multi-monitor setups and/or extremely large screens, and lets you keep an eye on windows that are currently important.

Our other MATE

While Cinnamon is steaming ahead with big changes, the same thing can't be said for MATE in this release. 1.6 from April is still in use, which is still a great desktop environment but it's being slightly shown up by the rest of the distro. The team has been concentrating more on getting it stable on other distros, which has really been the main issue with MATE since its inception. While it's doing that well, it means it's not receiving much innovation for the time-being.

This is sort of indicative of Linux Mint 16 in general. When we spoke to Clem Lefebvre around the release of Linux Mint 15, he mentioned that 16 would be a 'harder sell'

as it wasn't receiving the wealth of great new features as 15 was. However, it's important that they release it like this now so that when Ubuntu 14.04 LTS is released, the crowdsourced bug testing of a major release like this will iron out any minor issues still remaining.

The big question, though: is this still the fantastic Linux Mint distro we've come to expect? The answer is a resounding yes – while very little is new, with such a big change it's much more important that they were able to at least replicate Linux Mint 15. They've managed to do that and a little more, with the result being a great Linux distro that, while great for a home office, can be used anywhere.



We couldn't have asked much more of Linux Mint for this release, but it's managed to meet all our expectations and slightly exceed them. An important release and frankly still a great distro, politics or otherwise.

GNOME is not for everyone; luckily, Fedora offers more options as well



The search function is great for finding apps and documents alike

Settings are presented separately in the search results, making for quick access to some options

Pros
The new installer is a bit better than previous ones and the new 3D printing tools are a nice addition

Cons
Still tripping over some of the more controversial, cutting-edge features that have caused issues

Fedora 19 Schrödinger's Cat

After a long-delayed and divisive Fedora 18, how has the latest edition of Fedora shaped up?

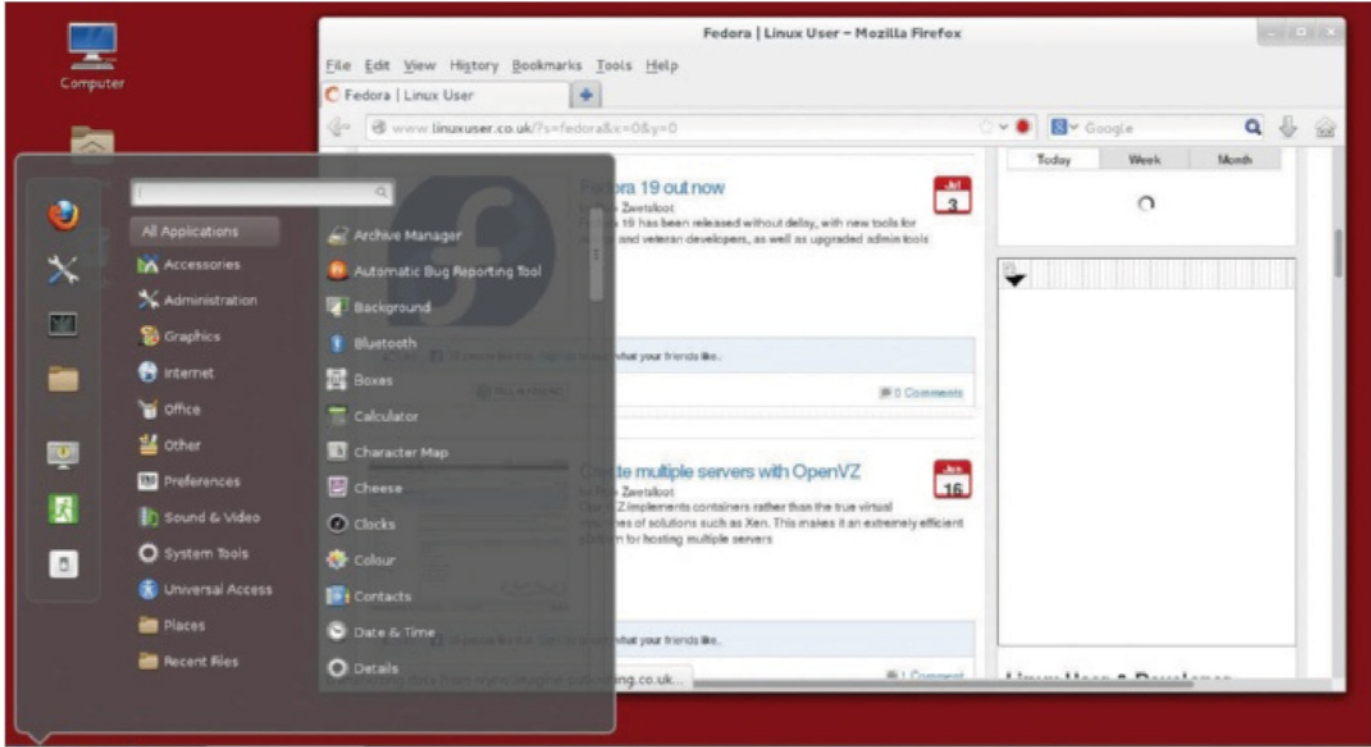
The famous Schrödinger's cat experiment is one of those stories from history that is perceived incorrectly in popular culture – like King Cnut arrogantly trying to stop the tide, or Bill Gates saying that 640K would be enough for everyone. Erwin Schrödinger's hypothetical experiment was actually a way of explaining how some interpretations of quantum mechanics were a contradiction of common sense. While this name was voted on for Fedora 19 by the masses of the internet, it's sort of indicative of the kind of problems people have been having with the default state of the distro for the last few iterations. GNOME has been moving quickly away from the traditional desktop metaphor

for years, with recent updates going against a mouse and keyboard workflow. The Anaconda installer update from Fedora 18 limited some options in favour of a more aesthetically pleasing experience. The distro has also not been particularly bug free, with systemd causing headaches for some. So, with Fedora 19 have some of these immediate issues been addressed, or are there new ones to throw on the list?

Upgraded installer

The first thing you'll experience with Fedora 19 is the installer, which has been upgraded again. Hardware recognition seems fine and there's

now a lot more control over the partitioning and editing of storage locations, an issue a lot of people had with Fedora 18. However, the method of doing so is not the most straightforward. Like in other graphical installers, you can select the hard drive you wish to use; however, instead of then performing a manual partition, or selecting a recommended installation scheme, you need to start 'reclaiming' space. This can be done by either completely deleting any existing partitions, resizing, or creating your own through the reclaim option, otherwise it will automatically try to fill the space already made. Pre-existing swap partitions are ignored, for some reason, and 19 will create its own if space



“There’s now a lot more control over the partitioning and editing of storage locations”

is cleared out. The installation will start before you can finish creating a root password or user, saving some time, but it still seems that this new installer is not ready and needs a lot more time in the oven.

GNOME 3.8

If perhaps the installer is supposed to be more in line with the simplification of GNOME, it’s doing a good job. GNOME 3.8 hasn’t had many major changes over 3.6, insomuch that it’s still ‘dumbed down’ in many respects. As if to highlight that this is the path the GNOME project is taking, a video explaining how to use GNOME launches on first boot. Credit where it’s due, though – the search function launched from whatever the Windows key is being called these days has always been good. Even if it’s

supposed to be a substitute for a large amount of a workflow, the search function part is often faster than mousing around and has now been upgraded to include some system settings results in your search. Sort of like a hybrid between the same functionality in Unity’s HUD and the classic search results between them, or the inclusion of Amazon adverts.

Stay up to date

The software updater has also been separated from the generically termed ‘Software’ package manager now, as well as in the applications list, although it’s still accessible from there. It’s here and in the repos that you can access all the alternative desktops, although there are three extra spins of Fedora that you can also use. As well as the KDE one, there’s the lightweight Xfce and LXDE choices, with other popular desktops such as Cinnamon 1.8 and MATE available in the repos. This version of Cinnamon is built to work on GNOME 3.8, so you won’t need to downgrade.

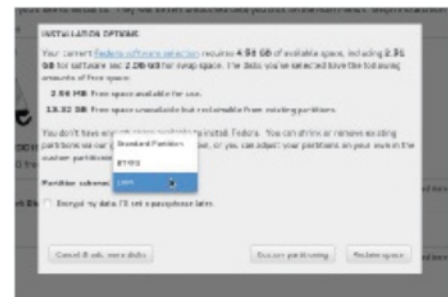
The distro itself is more stable than Fedora 18 on our physical setup. In a virtual machine, though, we experienced some noticeable slowdown and minor graphical glitches – so for virtual distribution you may need to do testing before deployment. Fedora, then, is not quite

the beast it used to be, with its cutting-edge stance harming it more than it has in the past. For those that were using Fedora 18 without issues, it’s a great upgrade; however, for those that moved away in recent years, this won’t bring you back. The box contains only one quantum waveform – and it’s not looking good for the cat.



The latest version of Fedora has fixed some of the problems we experienced with the previous editions. However, there’s still a way to go for some of its features.

- The installer has been a sore spot for Fedora users since 18 and while the new one takes some steps to fix the issues, it still needs development





■ The Control Center is a great way to use Mageia

Mageia 3 RC

Delayed at the last second due to installation issues, is the rest of Mageia 3 working fine or is there more to the delay?

Pros

A smooth distro with plenty of choice and customisability put up front for the user that's easily understandable

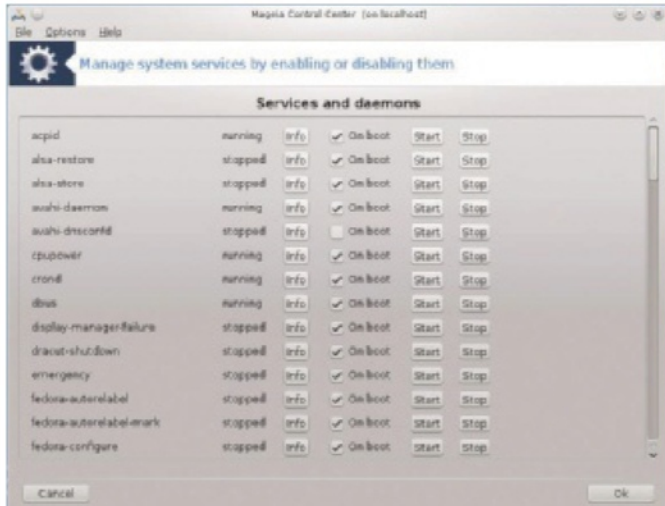
Cons

No live environment on the DVD version, and problems with setting up the correct repositories after install

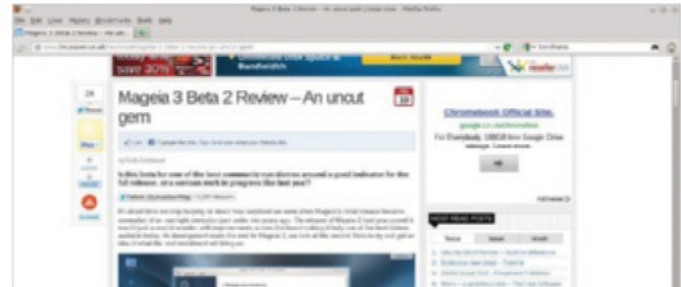
We've been looking forward to the newest Mageia for a while now, especially after how much we liked Mageia 2 last year. It fixed a lot of the teething issues you'd expect with a newer distro, although there were still a couple of small issues we had at the time. Mageia 3's development looked set to be the best version of the distro yet; but, due to blocker bugs with the release candidate, the final version has been delayed by a couple of weeks. Unfortunately, this means we were unable to review the final version before we went to print – however, the blocker issues don't affect the main Mageia experience.

The bugs themselves are related to the installation, although they're different issues to what we experienced in the beta builds. Hardware detection hampered us during the review of the release candidate, but this will be fixed before the final release. Otherwise installation is very nice, offering a straightforward run-through of all the necessary options, with plenty of advanced tinkering to be done.

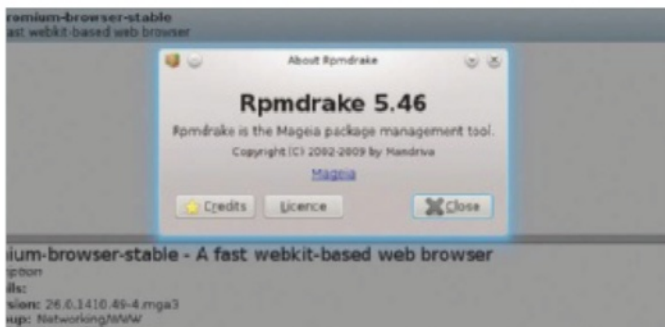
The full DVD version gives you a variety of desktop environments and package types to install – and even with such a large amount of software to install, it doesn't take too long on modern hardware. You have the choice of lighter



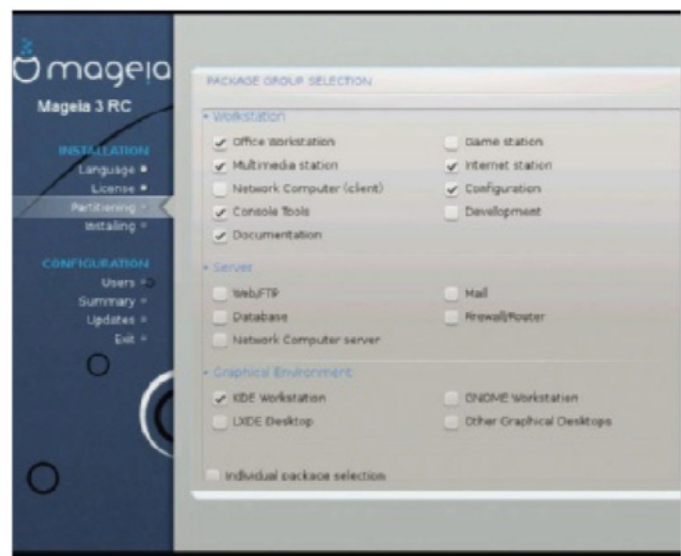
■ Even intermediate users can access more advanced features



■ It's much better than the beta



■ Make sure you set up the repos correctly for packages



■ The installer is very comprehensive, and easy to understand

desktops as well in the DVD version, in case you're trying it on a slower or older machine.

With all the defaults, Mageia 3 drops you into KDE, albeit the toned down version without all the graphical effects. The launcher menu is still present, but there's no searching on it. The most important applications are provided as quicklaunch icons on the taskbar, including the all-powerful Mageia Control Center. We've talked about the Control Center before, but it's worth repeating – it's a fantastic graphical hub with access to many settings and features of the distro and hardware, allowing you to manipulate them from a series of homogenous menus that then go straight back to the Center itself. For those wanting to use command-line tools or other graphical tools, they're still available, but it's nice to have all these options up front.

There are a huge number of default applications as well from the DVD install, with a selection of office tools, image editors, media players, media editors and internet tools to amply get you started. Of course, you can augment and

“With all the defaults, Mageia 3 drops you into KDE, albeit the toned down version without the graphical effects”

change these via the repos. Mageia otherwise ran very smoothly and was fairly quick, even without resorting to using a light desktop environment. The only issue we ran into was the recurring problem of the distro not setting up the online repositories after installation, although you can opt to search online for any updates during the installation. It's actually easier than before to activate the online repos; however, packages on the disc don't always have the necessary dependencies available, so it's a necessity.

Overall, we're extremely pleased with the way Mageia 3 has turned out – it has a great selection of distro types to install from, no specific allegiances to any desktop environments or software, and an

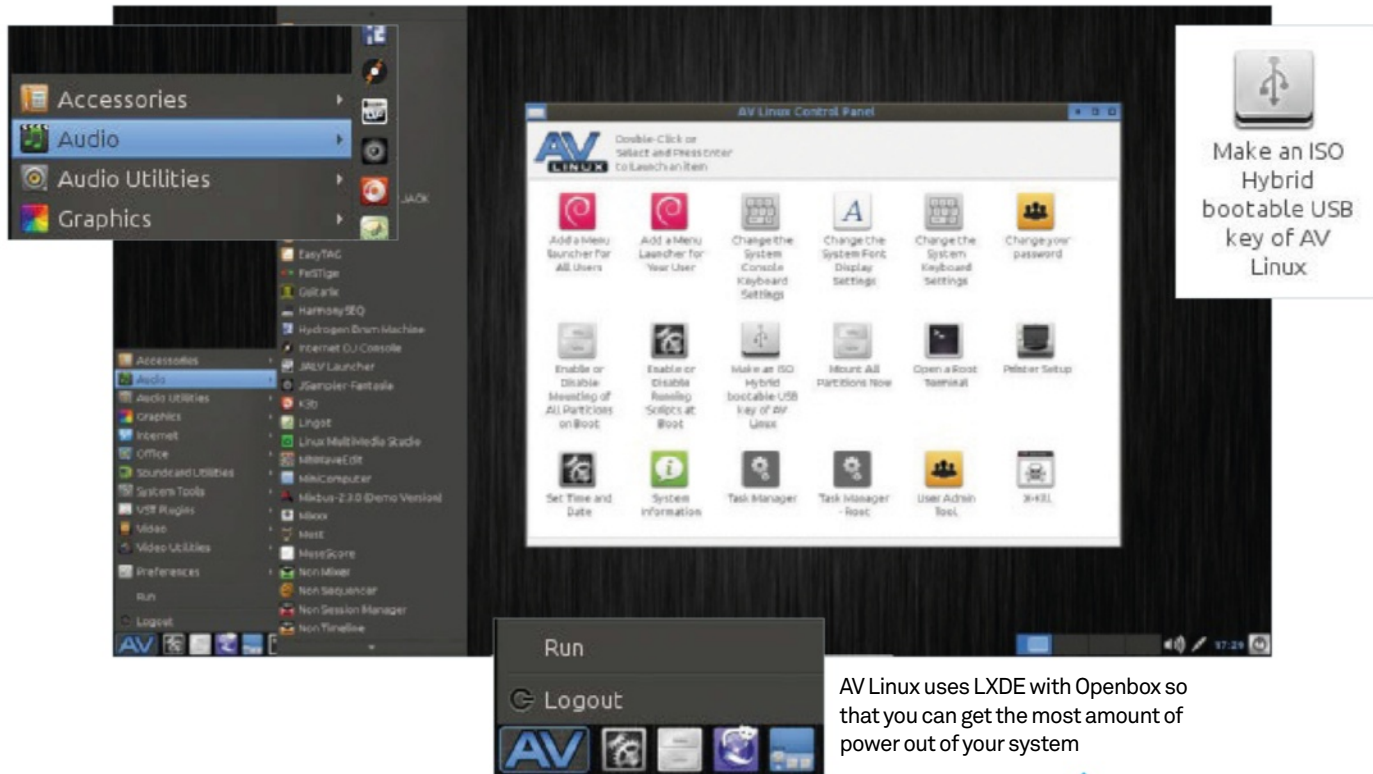
approach to the custom applications that allows for people of intermediate skill level to do fairly advanced Linux tasks. It's great to use, work and develop on, and may well be the perfect version of Mageia – a distro which has now been with us for two years.



If you've been one of the many that jumped on board with Mageia early, this is pretty much the perfect version of the already great distro. People who are looking to switch or are on the fence have no excuse not to try it out – it's fast, stable and an all-round pleasure to use in whatever you do.

Utilities have been split up in the menu now, allowing for easier access to the hardware tools

The AV Linux Control Panel allows for a lot of customisation of the distro, including creating a special Hybrid ISO for USB



AV Linux 6.0.1

- Pros**
A real-time kernel for reduced audio latency, and every major media editing application available as default
- Cons**
The installer is a bit basic, and the real-time kernel can be a resource hog if you don't specifically need it

A highly customised Debian designed for video and audio professionals, how exactly does it differ from other distros? And does it deliver?

AV Linux 6.0 came out towards the end of last year after an apparently difficult development period. AV Linux does take a while to iterate, though, and there were some big changes from version 5 to 6 – such as switching from Linux kernel 2.x to 3.x for better hardware support, and some major updates to all the packages. A point update has been released for 6.0 now, which includes some minor upgrades and bug fixes throughout the distro and its apps.

AV Linux comes as a fully functional, live-booting OS, perfect for writing onto a USB stick

or live CD and using it where you can. It does also include a fairly basic installation function, which gives you just enough control to partition your system and then install the distro to it. While it isn't as pretty or user-friendly as some of the major distros, it does the job and explains what you need to be doing at each of the steps. This includes giving instructions on how it wants the hard drive partitioned before letting you launch into GParted to do it. The biggest hassle during installation in general is changing locale and keyboard if you're not American.

Live performance

One of the things that make it great as a live distro is its many extra drivers for a lot of proprietary audio and video hardware, such as sound cards, graphics cards, MIDI controllers and more. These are all carried over through to the installed version and you can then start customising which versions of the drivers you want to use, and keep them. This allows you to get the most out of your system for AV work without having to set it up every time. There are also a lot of sound-card utilities that allow you

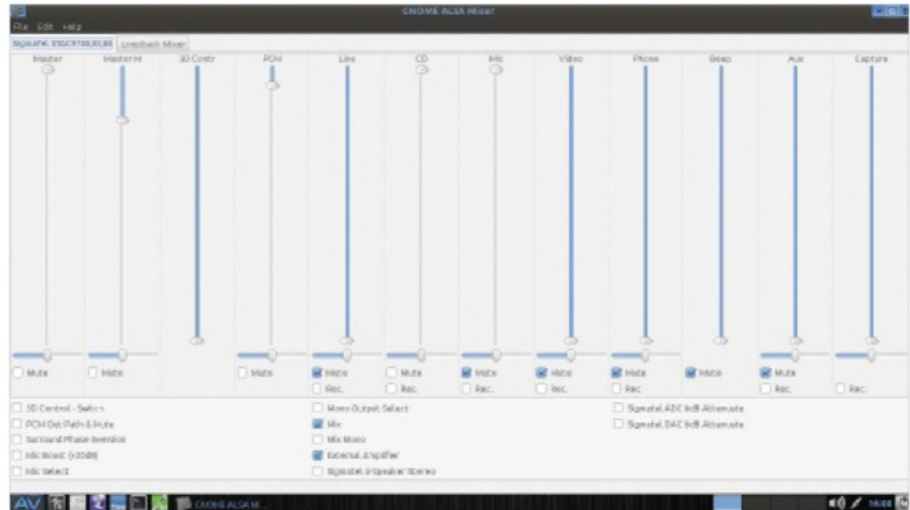
“The real-time kernel allows for a much lower latency while recording audio”

to edit the levels through a variety of different applications with different effects to the sound input, although it seems to prefer you use one at a time rather than a combination.

Speaking of sound input, AV Linux’s real-time kernel is one of the distro’s best features for professional audio engineers. The real-time tag on boot allows for a much lower latency while recording audio, keeping things a lot more accurate than the standard Linux kernel. You can remove this if needed, though, as the real-time kernel does take up a few more resources than usual, noticeably slowing down our test machine while it was activated.

Modifying defaults

There are several boot-time cheat codes that you can apply on boot, one of which is the `-rt` option that enables the real-time kernel. By default, there are options to enable threaded IRQs, which aids in the audio latency, as well turning off the hugepages memory management feature. Disabling hugepages is apparently another step in keeping the latency low, but it causes



■ The selection of sound mixers and audio drivers don’t work in harmony, but they offer a lot of control over a sound card when you need it the most

memory-heavy applications such as video editors to not work as smoothly. There’s also an option known as `noautogroup` that allows for the desktop to work a little better, even when CPU-intensive operations are under way.

One of the upsides of AV Linux is the sheer amount of audio, image and video editing software pre-installed onto the distro. Mainstays such as GIMP, Audacity and Cinelerra are joined by 3D modellers like Blender, and other video editors for different skill levels such as OpenShot and Kdenlive. The full LibreOffice suite is also available in case you need to write or present something, and of course a full

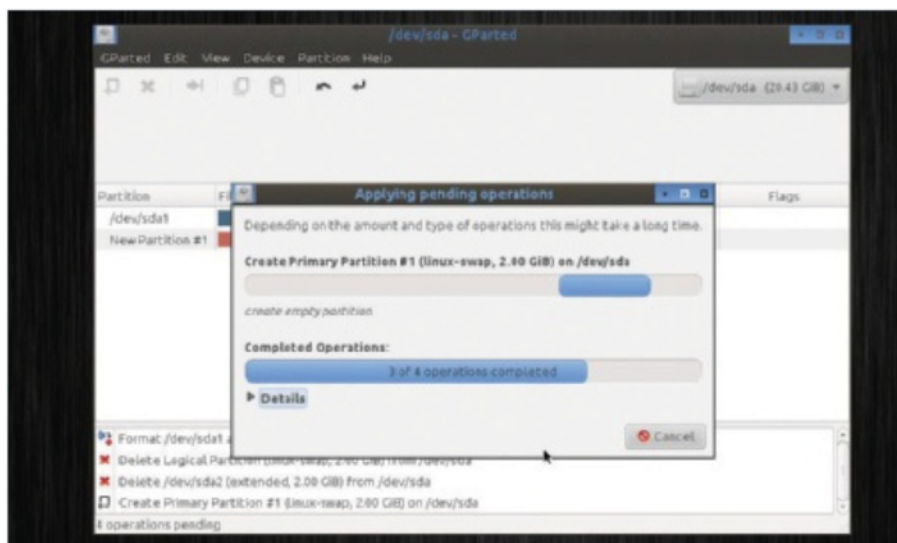
complement of internet browsers, messaging clients and media players are included to round out the experience. All this is especially helpful for the live version of the distro, as it’s all there without preconfiguring, and there’s a lot of extra utilities and tools that would take a long time to set up and customise to this extent. However, you can make a hybrid bootable USB key using current system settings to create a more personalised version of the live distro, with software installed or removed using the standard Debian packages it’s based on.

All inclusive

It’s generally a fantastic editing suite. The use of LXDE and its policies on screensavers means it can eke every bit of power from your system to make sure latency and rendering are the best possible. And on top of that, it gives you a huge amount of control over the way the hardware interacts with the distro to optimise the system even further. It does use a lot of non-free software to achieve this, though – unfortunately this is more of a problem with the entertainment industry than anything else, but allows people to do the kind of work they want to use AV Linux for.



Possibly the perfect audio editing suite based on Linux, especially for one that’s ready out of the box so to speak. The real-time kernel option is a great feature for sound engineers, reducing audio latency, and there’s a lot of driver and hardware control for everyone else.



■ Installation is more manual than some distros, including using GParted to set up your hard drive. The process has clear and concise instructions, though



■ Bodhi looks and works like any standard desktop distro

Bodhi Linux

Pros Extremely lightweight and customisable, and doesn't sacrifice anything over standard Ubuntu to achieve this either	Cons No real extras other than the inclusion of Enlightenment, therefore no extra repos with packages missing from Ubuntu
---	---

Using the Enlightenment desktop environment, Bodhi is an incredibly lightweight and highly customisable distro based on Ubuntu

We're constantly seeing Ubuntu being used as a base for other distros, whether it's official respins with a different desktop featured, more involved remakes such as Linux Mint that 'fixes' the problems with Ubuntu, or lightweight/green versions of Canonical's distro like wattOS. Bodhi is on the lightweight end of the scale, taking it the extreme by using Enlightenment as the desktop environment, and also including very few default apps as standard.

Showing off its roots, initial interactions with Bodhi are very familiar to anyone who has

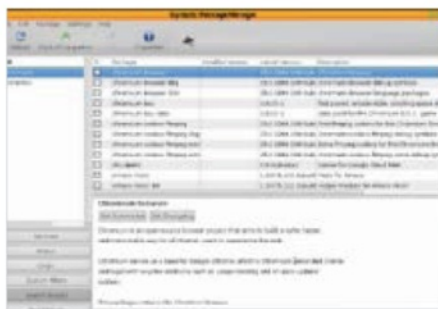
previously used Ubuntu. The installer has the exact same simple layout and options, including the installation while setting up users, and downloading updates in the process as well. The live disc gives you a few booting options too, with safe graphics modes or the ability to load into RAM so it can be run on older or slower systems.

Installation is fairly quick and it takes up very little space on the hard drive initially.

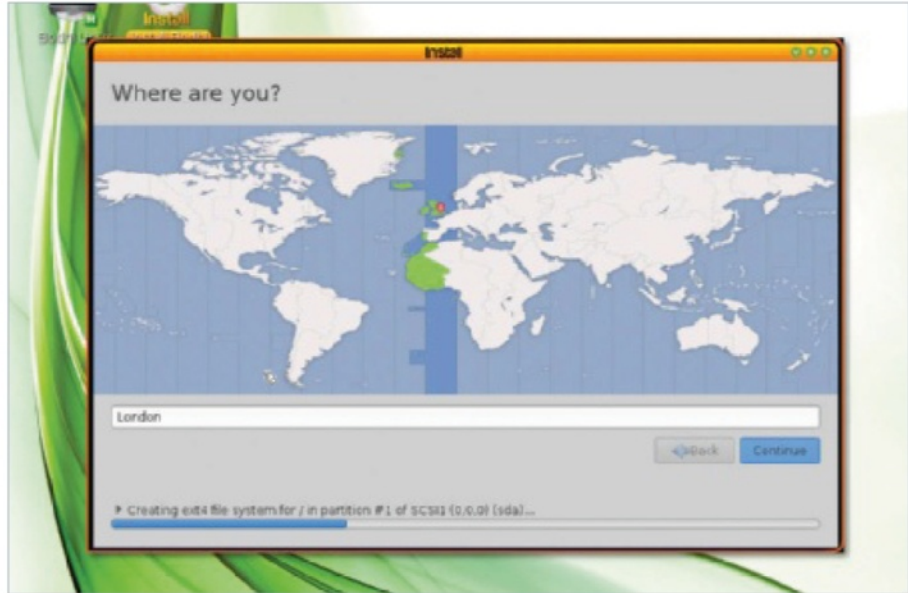
On first boot, you'll be asked to set your E17 profile out of a small selection of desktop configurations. There's a bare setting, with only the essential desktop menu to access everything, and other settings for various desktop elements.

“Initial interactions will be very familiar to anyone who has used Ubuntu”

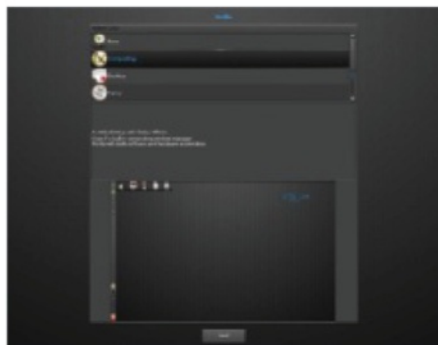
“Boot time is incredibly quick, getting from POST to the login screen within seconds”



■ Bodhi is powered by Ubuntu 12.04, with the same wealth of packages as well



■ The installer shows off the Ubuntu roots



■ Enlightenment is highly configurable with different profiles



■ The start menu hierarchy can get a little confusing

This gives you a great amount of choice on a sliding scale of usability to sheer speed, even being able to add in advanced effects and such. This can be changed any time afterwards and just acts as your default selection for the time-being. You can also choose your theme, but that has less of an impact outside of aesthetics. After the first-time setup, boot time is incredibly quick, getting from POST to the login screen within seconds.

Enlightenment itself is an interesting experience, parts of it being very similar to KDE or Cinnamon and the like, but with a somewhat complicated start menu system. It takes a little getting used to the hierarchy and can make tweaking the distro on first use a bit infuriating. However, for day-to-day usage of just installing and using applications or performing tasks in the terminal, it doesn't get in the way too much.

This menu can also be accessed by clicking anywhere on the desktop, which is a nice touch when it would otherwise not do anything. It has all the amenities of something like KDE in the traditional desktop mode, with virtual desktops, quick launch buttons, a list of open windows etc.

Packages are maintained with the old faithful Synaptic package manager. It links into the standard Ubuntu packages, with very little else added from any Bodhi sources. This also means that any repos or PPAs available for Ubuntu 12.04 will work in Bodhi, expanding an already extensive list of apps. Coming with the system by default is very little of note: text editor, terminal emulator and Midori for web browsing. All of this can of course be extended, but it's a great lightweight start so you can make sure bloat is at a minimum.

Bodhi is a great distro, and a great idea. Using the LTS release of Ubuntu as its base, it's taken a well-supported and pretty stable distribution and created an incredibly speedy and lightweight version that is perfect to get the most out of computers, new and old. Enlightenment is a great windows manager that rarely gets much attention, and that's completely undeserved. Try it as an alternative to wattOS.



A lightweight and lightning-fast distro based on the rock-solid Ubuntu 12.04, meaning it will be well supported for years to come. Enlightenment brings its own flair to Bodhi, making it small yet very usable without much or any compromise.

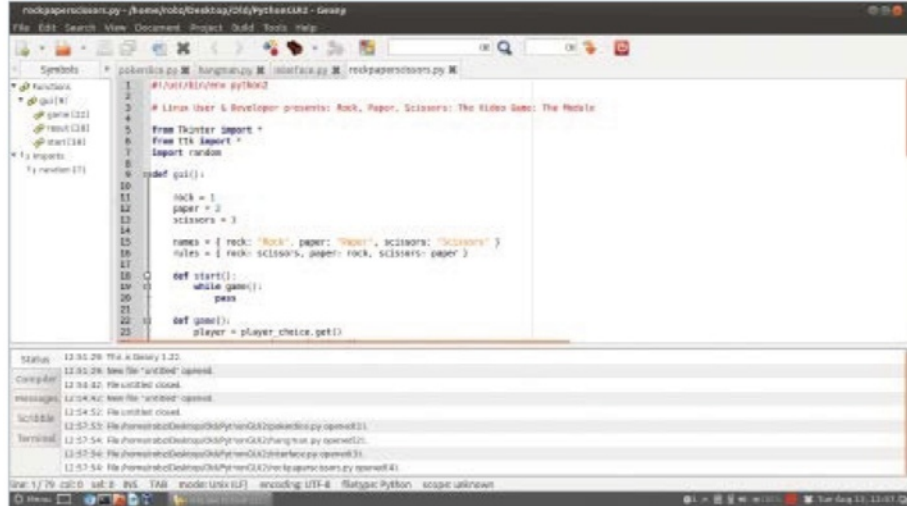
Geany

A fully featured IDE that's a little more lightweight than most

Geany is sometimes mistaken for a glorified text editor. In reality, it's a full IDE, albeit slightly more lightweight than most.

Geany has multi-language support out of the box, with easy options to create files that use different languages as part of the main interface. Projects are kept in specific folders, like Eclipse, although there is a file to go with them that Geany can read to manage the projects. Multiple projects can be viewed and edited at a time, all in different languages. There are also debuggers and builders for the various languages that support them, and you can run and test stuff like Python that doesn't need compiling as such.

The rest of the interface for Geany is clean and well labelled. Code is automatically highlighted with the correct syntax and there's a smart tab in the left column that allows you to track and view the different variables, functions and classes in the projects and code. Navigation through the code is fairly simple via context-sensitive menus and nice options within the rest of the interface, and the usual code-editing tools such as commenting out a selection or indenting are all present.



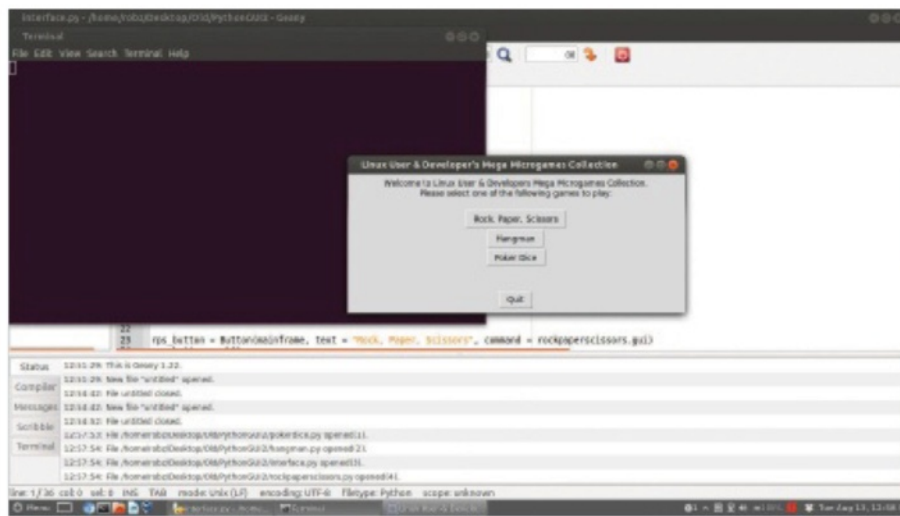
■ Geany's interface is smart and easily breaks down code for workflow purposes

“Code is automatically highlighted with the correct syntax”

Sadly, Geany's plug-in support is pretty dire. Some plug-ins extend the basic functionality for a few of the languages, but there's not the kind of depth as Netbeans or Eclipse in the sheer number of available plug-ins.

There are a lot of ways to customise Geany with the standard tools and menus, though. Almost every part of the interface and workflow is editable, with ways to even change the characters required for autocomplete suggestions.

Geany is a very smart IDE, with a low barrier to getting your project started straight away. It does lack in some of the features of some of the bigger IDEs, such as more advanced unit testing and debugging, although it will let you browse compile errors. The plug-in selection is also pretty poor, so it may not be extensible for specific functions.



■ The native languages are supported quite well with compilers and such

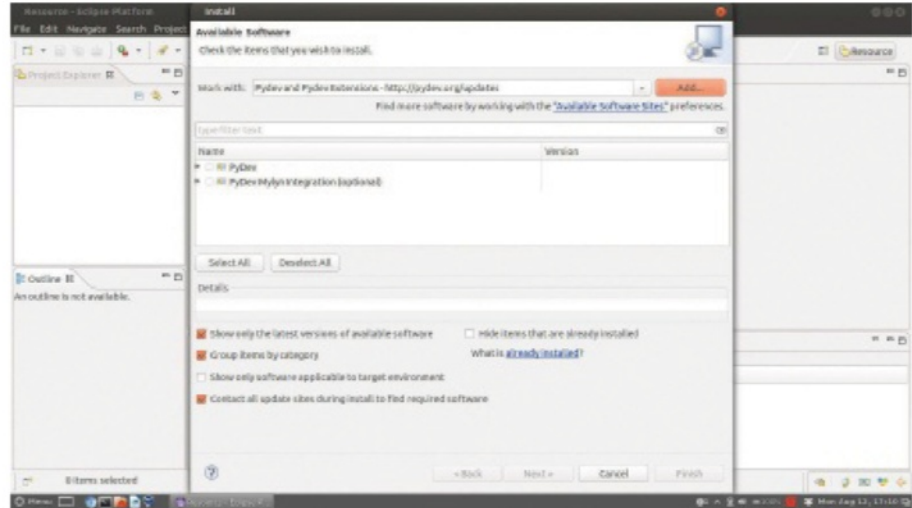
SCORES		
Installation	Readily available in most repos and requires minimal dependencies	9
Workflow	Workflow is superb for the variety of different languages included	9
Features	No proper debugger, but a great selection of build and project management functions	7
Plug-in support	Although it doesn't quite need it like the others, there are very few plug-ins available	4
Overall	Geany is a great, lightweight IDE with good native support for a variety of languages, but there's not much room for expansion	8

Eclipse

The ubiquitous Eclipse is an industry standard – how does it fare against more community-run efforts?

Eclipse is one of the most popular development suites around, and at first glance it's easy to see why. While created mainly for Java development, Eclipse is highly customisable through plug-ins. Thanks to its popularity and community, this has resulted in a great selection of add-ons that enables Eclipse to work with just about any language. These plug-ins allow for more than that, with a marketplace full of interface and behavioural modifications alongside the language elements.

Eclipse has great project management tools as well, with a smart tabbed interface, and plug-in-specific menu entries for starting projects in different IDEs if needed. Projects are located in plain folders in the location of your choice, allowing for easy access of source code if you need it, instead of being inside a proprietary file. The function search ability works well and the interface has plenty of tips and warnings for



■ Plug-in support for Eclipse is top-notch

anything that might be inefficient in terms of the code. There are plenty of editing tools too, such as simple indent or dedent options.

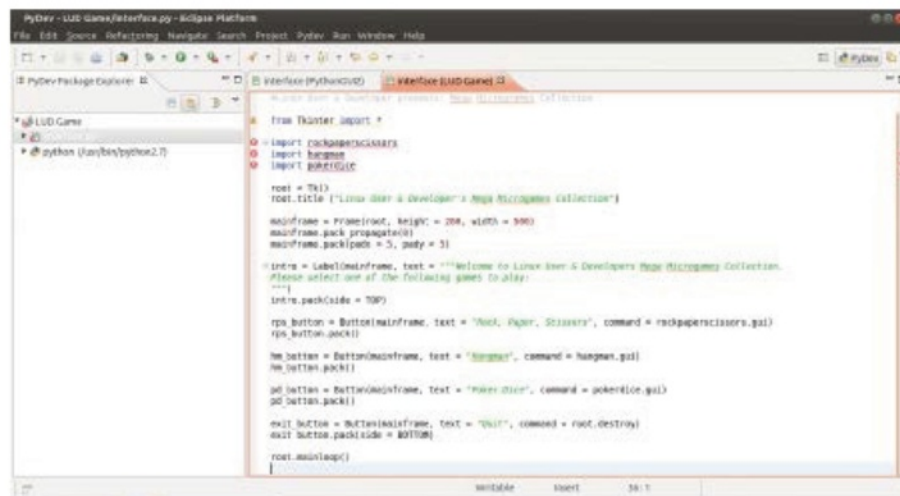
The debug suite in Eclipse is fully featured, with various ways to run, check and unit-test code, although this depends on the plug-ins to some degree. The tools are there, though, and most of the major plug-ins seem to use them.

The plug-ins are handled by a repository system, which lets you keep any add-ons up to date. While there aren't a huge amount available by default, it's easy enough to add more to the plug-in manager and you can even

select which extensions to install from each of the repositories.

Eclipse is customisable in other regards, with an expansive properties and settings menu that lets you edit a huge amount, from the way patches are viewed to little things like key bindings and other shortcuts. Eclipse is a fairly big suite of packages, though, and easily the biggest resource hog out of all of the IDEs we're testing. It does have probably the best extensibility of all these IDEs, however, meaning it also probably has the most to offer those who work on a lot of differing projects.

“A great selection of add-ons enables it to work with just about any language”



■ Code navigation and highlighting aids your workflow

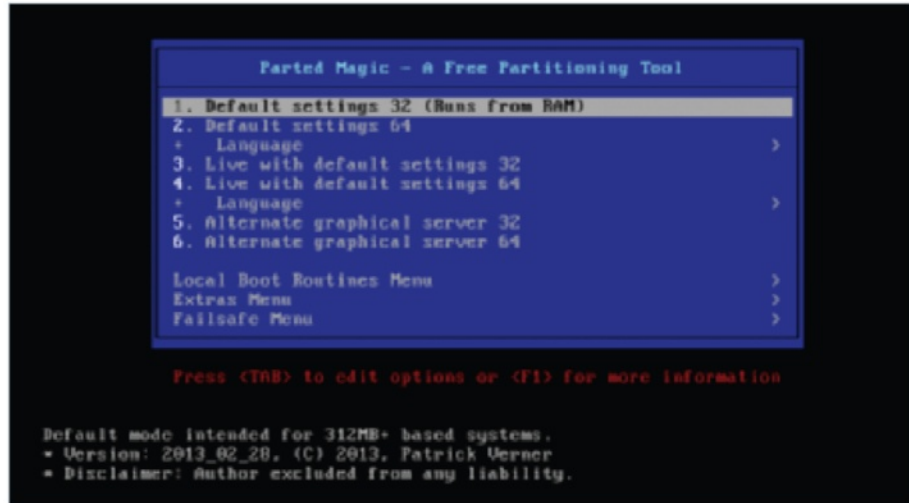
SCORES		
Installation	Eclipse requires a lot of dependencies; however, it's available in most major repos	7
Workflow	Smart interface design that lets you easily navigate projects and code	9
Features	A great set of features by default, although mainly for handling Java	8
Plug-in support	The best support around, with a great repo system offering a wealth of add-ons	10
Overall	Eclipse is popular for a reason – its got a great selection of features that are easily extensible to suit almost any need	9

Parted Magic

A good recovery distro with a focus on hard drive partitioning

Parted Magic gets its name from two of the partition editors featured in the software suite, which is an indicator of its focus. There's a lot of partition editing software on the disc, both in graphical and command-line form, and there are even tools and packages to help you diagnose the health of your hard drives. Shuffled in with these apps, though, are a few great tools for other system recovery and maintenance tasks.

For starters, ddrescue is included with the other partition and editing tools. This data recovery tool is very powerful for fixing file corruption and suchlike, with ways to compare it to previous versions, even if they too are corrupt. Parted Magic lists all available tools on its website, with categories and a small description for each, allowing you to look up



■ There's plenty of choice at boot time

the relevant package for the task you wish to accomplish. While this is great, there's otherwise very poor documentation on how to perform various maintenance and recovery tasks on the website, something that we discuss in our SystemRescueCD review over the page.

The ISO itself is pretty tiny at just over 300MB, and comes with multiple boot options at startup for 32- and 64-bit kernels, alternate graphical modes, and whether you want to load into RAM or not. This extra bit of choice gives it a lot more

compatibility on older or slower systems, and in general it's very fast to boot and load.

In addition, it has some other maintenance features, such as some rsync-based utilities for backups, Clonezilla for disc cloning, and even an anti-virus through ClamAV to scan your drives. There's also access to the internet through Firefox so that you can look up any information or tutorials for the tasks you're aiming to do.

Parted Magic is a good recovery distro, even if it's a bit more focused on hard drive partitioning. There's not much in the way of extra support for if you have it installed to a USB stick or something; however, there isn't much customisation you'd need to do to it between uses.

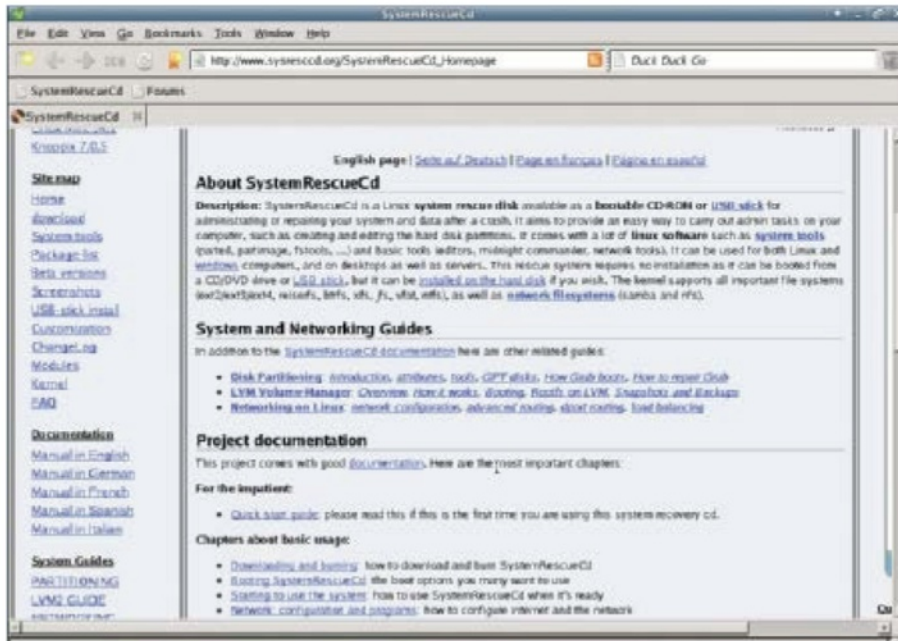


■ Parted Magic is full of graphical tools

SCORES		
Boot	Quick to boot, with some advanced boot options	7
Recovery	Using ddrescue, you can recover and fix files in a number of ways	7
Ease of use	A lot of graphical tools, although not much documentation on the website	8
Extras	Backups, disc cloning, system health and partitioning tools are also available	9
Overall	A great live distro to have to hand if you need to maintain a system, but not the best for recovery	7

SystemRescueCD

Great for recovery and general maintenance



■ The online manuals offer great guides

On first impressions, SystemRescueCD seems to tick all the boxes we'd want it to tick as a recovery distro. The live-booting OS comes in at a fairly small 370MB and immediately offers up a huge variety of boot options, with the choice of 32- and 64-bit, languages, alternate boot methods etc. This then quickly loads into a command-line interface for the interest of speed, and it either lets you continue doing work on the command line or gives you information on how to boot into a graphical environment. This environment is fast and gives you some extra graphical tools to do your recovery tasks.

All in all, that's a great start. It's got plenty of applications to help you with data recovery, including ddrescue that we mentioned earlier, which work mostly on the command line. However, the SystemRescueCD website has some great guides to get you going on that.

“A fantastic tool for performing recovery tasks”

This 'manual' is set as the homepage of the included web browser (Midori, to keep it fast and within Xfce) so as long as you have an internet connection, you can easily access it.

There are other maintenance apps as well, including partition editors and backup utilities,

SCORES		
Boot	A huge number of boot options available through a straightforward menu	10
Recovery	Multiple recovery tools that can be used in various ways	9
Ease of use	A lot of command-line tools, but they're thoroughly explained in the manual	9
Extras	Other system maintenance tools round out the selection of apps	9
Overall	SystemRescueCD is a fantastic tool for performing recovery tasks on your system, or doing some general maintenance	9

making it multifunctional as a recovery and maintenance distro. You can install it to a USB stick from inside the distro with its own tool, or do it via a Linux distro with a script included with the ISO. You can even build it yourself.

It's a well-executed concept that does just about everything we'd want a distro of this kind to do. You can quickly work from the command line if you need to, or boot into a graphical environment if that's how you prefer to work, and being able to easily clone itself from within the distro is incredibly useful as well. Having a stick or a CD lying around with SystemRescueCD on it could easily save your computer one day.



■ There's a wealth of boot options for different systems

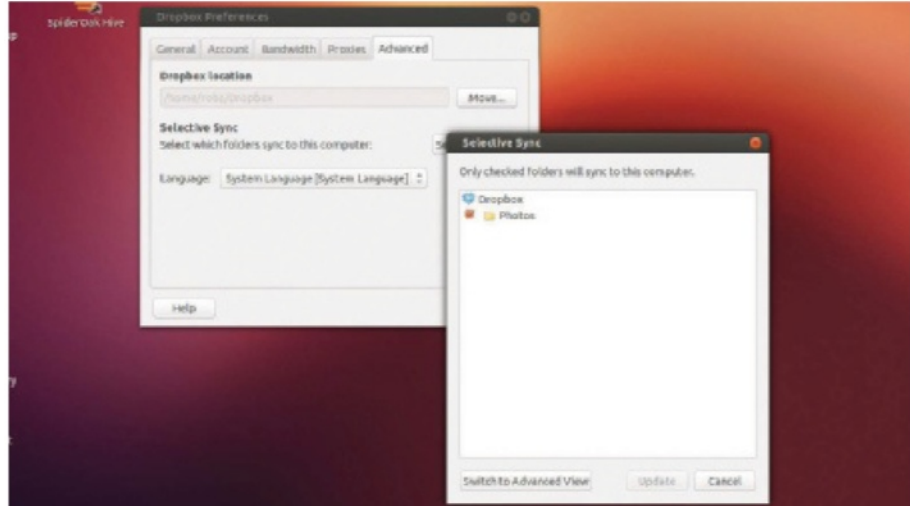
Dropbox

No introductions necessary for the king of cloud storage

Dropbox is synonymous with cloud storage; in fact, when describing other cloud services to less tech-savvy people, the conversation usually includes the phrase ‘it’s like Dropbox’. Popularity doesn’t necessarily mean quality, though, which is why we’re taking an in-depth look at its features to see if it deserves its reverence.

Here are the facts then – Dropbox is a cross-platform service offering 2GB of free data, with small increments available for referrals and such, which is upgradeable for a price. This starts at \$9.99 a month for 100GB, which seems to be somewhat of a standard in the cloud storage space and is cheaper than Ubuntu One’s 20GB add-on scheme. Dropbox’s Linux client is fairly simple compared to the other platforms it works on; however, it offers roughly the same level of functionality without some of the bells and whistles those versions have.

The client only allows for syncing of the main Dropbox folder, rather than letting you select different folders to also sync or back up.



■ Selectively sync specific folders in Dropbox

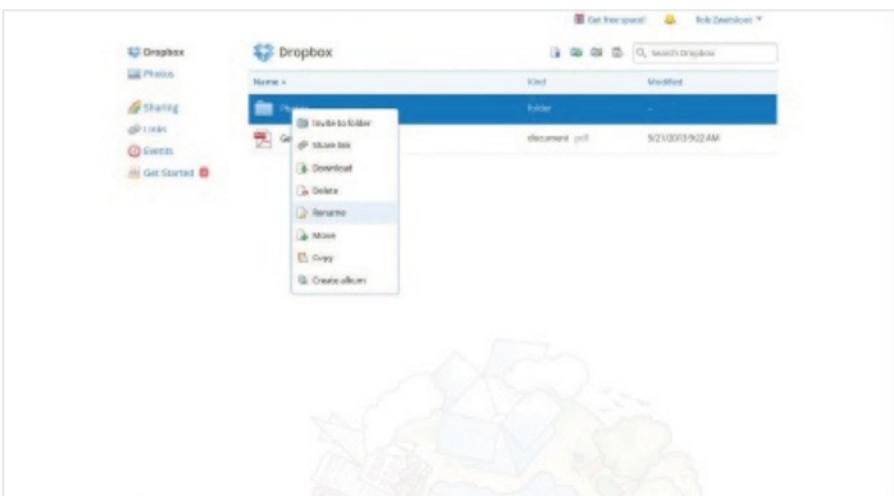
However, you can tell Dropbox to selectively sync specific folders from the main Dropbox directory, just in case there are files that only need to be accessed by specific devices, or just via the web interface. You can also share folders between other users, allowing you to collaborate on work, or just quickly and easily transfer files between each other. If you also have another client on the same network, it will allow you to transfer files over the LAN automatically – not a huge feature, but good nonetheless in certain situations.

The web interface for Dropbox is also one of the best, allowing you to easily navigate, edit and

change settings on files on your account and in any shared folders that you have ownership of. There’s also a pretty advanced undelete function that lets you browse the last few weeks of files you’ve deleted and restore them as long as you have space in your account.

The good thing about Dropbox is that it’s available on all desktop and mobile platforms, and it works pretty much seamlessly between all of them. While it doesn’t offer all that much space for free, it does have reasonably priced upgrade options and is one of the most stable services available.

“Dropbox’s Linux client is fairly simple compared to other platforms”



■ Have more control over your files via the web interface

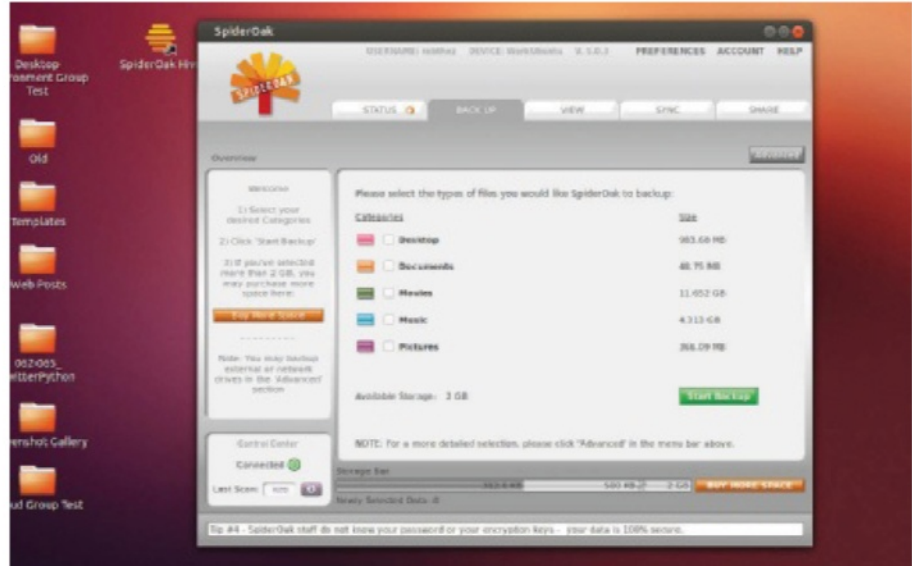
SCORES		
Space	Not a lot of storage space given away for free, but very reasonable upgrade fees	6
Integration	Integrates throughout a network and has a useful notification icon	8
Features	A good set of syncing features and sharing, but no extra folder backup options	7
Customisation	There’s actually not a huge amount to change in Dropbox, although there are proxy settings	7
Overall	Dropbox is very good and deserves its reputation. It’s solid, reasonably priced and works very well and very quickly	8

SpiderOak

A veteran in Linux cloud storage

One of the first commercial cloud storage services to make its way to Linux, SpiderOak has been around for a few years. This means a couple of things right off the bat – firstly, it has the same 2GB free, \$9.99 for 100GB storage deal as Dropbox. Secondly, it also has probably the best and most mature client program out of any in this test, which we'll shortly go into more detail about. One of the benefits of using SpiderOak is that the firm behind it is extremely confident in the security and privacy of your data on its server. It operates what it calls a Zero Knowledge scheme, where the server doesn't actually know what you have stored on it because it's encrypted. This is a good idea and should guard against most types of intrusion.

As mentioned, the client for SpiderOak is extremely good on Linux. The main interface supplies a lot of information on the current status of backup and syncing, what the current queue of uploading or downloading files is, a log of any changes made etc. Like Ubuntu One, you can select other folders to back up, as well as having a SpiderOak Hive folder that syncs between all devices. You can also heavily customise the way the client works – such as telling it to only back up files of specific size range, age range or excluding specific keywords;



■ You get complete control over the client and how it syncs

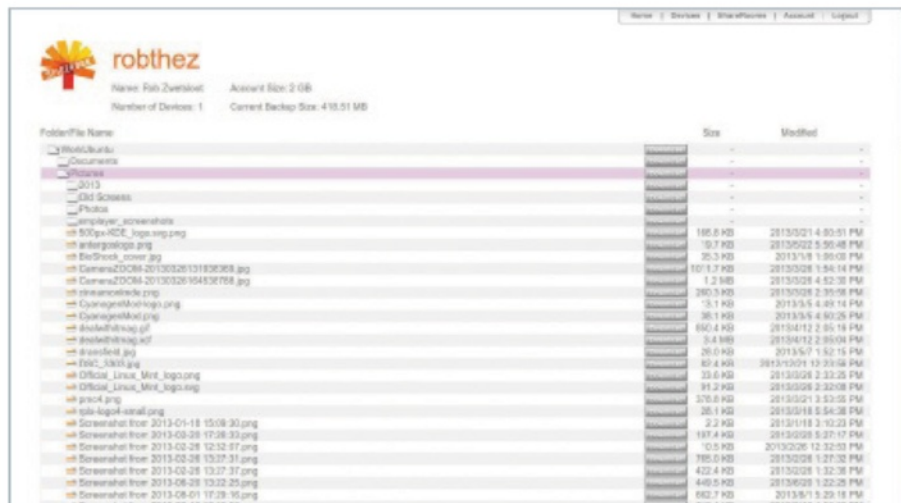
“The server doesn't know what you've stored on it because it's encrypted”

setting a backup and syncing schedule in case you don't want it to do it all the time; and it also includes LAN sync like Dropbox.

While the website end isn't as advanced as Dropbox's, there is an Android app which allows you to access your files more easily on the go. One of the neat functions included with SpiderOak, though, is the ability to sync two folders between devices, or on the same device, without having to use the dedicated syncing

folder. This is particularly useful for syncing between folders on the same device if you need some kind of a backup, or multiple users are working on the same system.

Overall, SpiderOak is an extremely mature cloud storage service – and unlike other solutions, it hasn't skimped on the Linux support.



■ The web interface is basic but usable

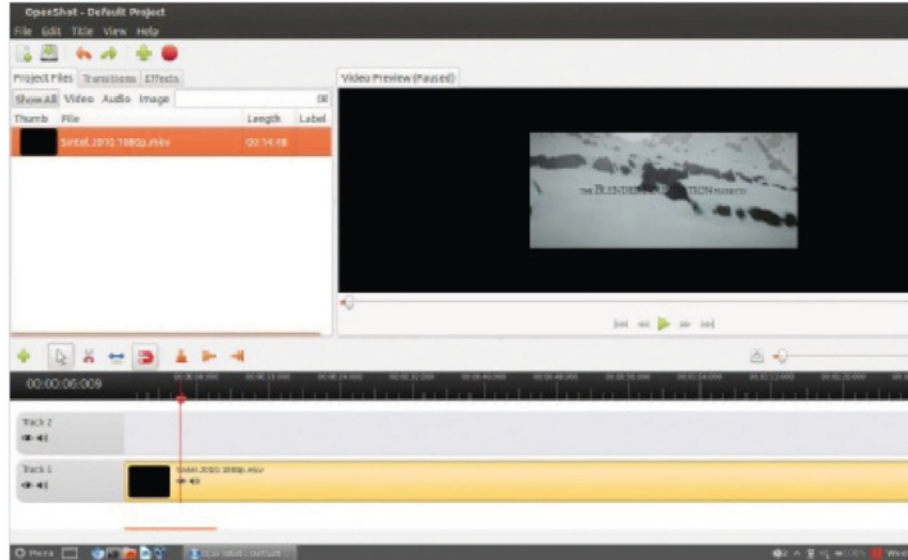
SCORES		
Space	Like Dropbox it only offers 2GB of free storage, but there are decently priced upgrades	6
Integration	SpiderOak is able to integrate well with a number of setups	8
Features	It offers the highest number of features we've seen for a cloud storage service	9
Customisation	Just about every aspect of the client is customisable with useful functions that can be enabled or disabled	9
Overall	SpiderOak is probably the best commercially available storage solution on Linux, and we don't say that lightly	9

OpenShot

An intuitive yet professional movie-making option

Built upon its own media framework, OpenShot is a fairly all-encompassing package that has similar codec support to FFmpeg. It's one of the newer entries in this group test, with its first release just over four years old now, and it's definitely one of the better editing suites we've come across. Like PiTiVi, the interface is nice and straightforward; however, unlike PiTiVi, this interface makes it easier to access a much deeper library of features and editing options.

The timeline consists of a track hierarchy, with higher tracks generally having dominance over the lower tracks in terms of what is previewed and encoded. There's no differentiation between music and video tracks, but it's smart enough to know not to cut off video if an audio track is placed higher than a video track. These tracks can be moved between by the use of transitions, which are visually represented very nicely with an arrow to determine the direction of play, bridging the tracks together to give a nice sense of the flow of the video itself. There are plenty of transitions available, ranging from dissolves,



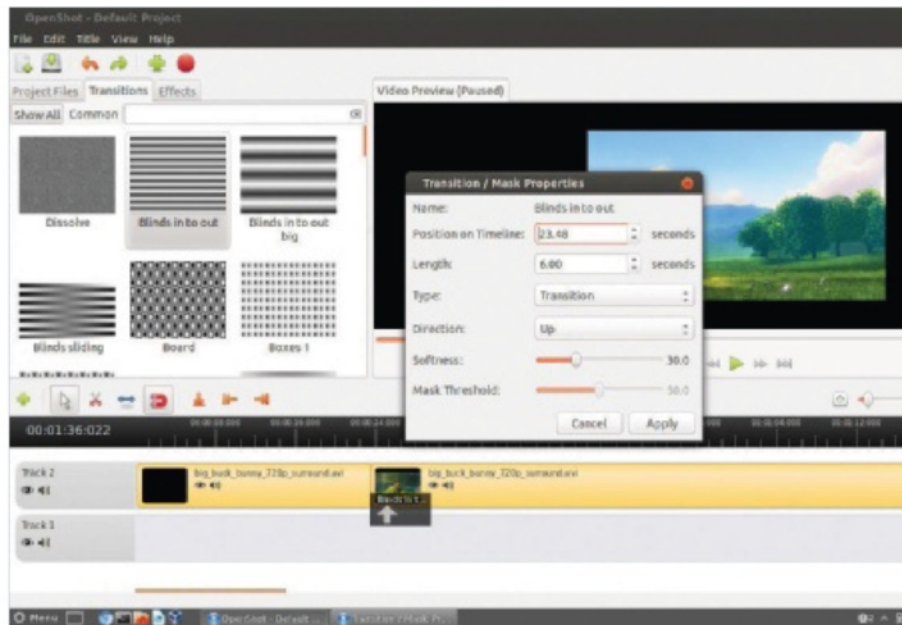
■ The layout is clean yet hides a lot of depth

wipes and fades to either emulating a specific style or even creating some impromptu special effects. There are some visual and audio effects available as well, which are added to the entirety of the clip in the timeline rather than a selected area. In this case you'll need to split up tracks to break up the effects. You can split off the audio from a video track as well if you so wish.

There are a fair number of rendering options, with plenty of presets and more advanced customisation available for bitrate, file format,

aspect ratio, quality and more, so you can create your perfect video file. This goes hand in hand with OpenShot's compatibility with a wide range of codecs: it's able to import from MKV containers as well as lots of other file formats and codec types.

OpenShot's slightly more advanced workflow over PiTiVi is a fantastic yet still easy-to-understand addition, which allows people to easily make much more advanced video clips and movies. Overall, OpenShot manages to combine ease of use with a good feature set.



■ There are a great selection of effects and transitions that are fully customisable

SCORES

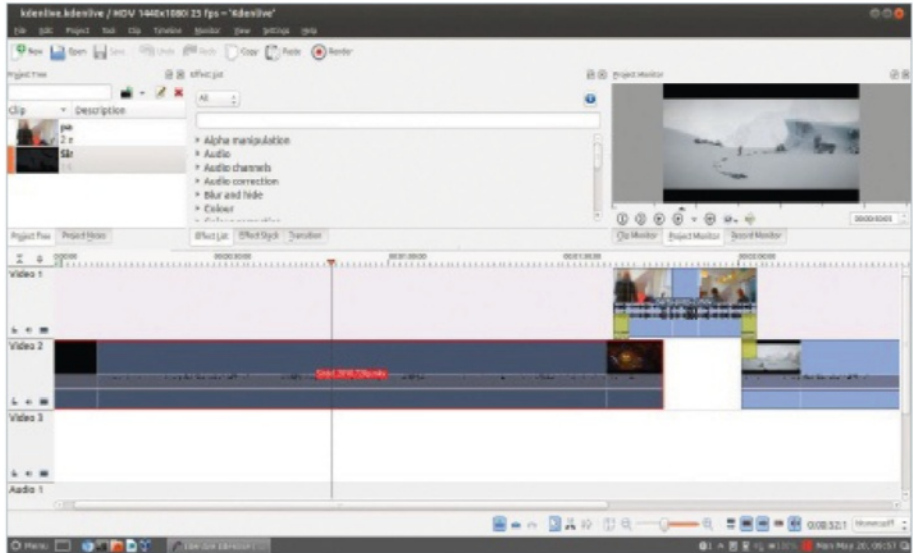
Installation	Widely available in repos, and has its own media framework	9
Ease of use	One of the easiest editors to use in this test, without dumbing down the features	9
Features	A great selection of video editing tools and tricks, as well as effects and transitions	8
Codec support	We found few issues importing video, and there were many export options	8
Overall	A prosumer-level video editor that is only slightly more difficult to use than your basic video editor	8

Kdenlive

A full-featured video editor – it’s the complete package

Whereas OpenShot is a straightforward video editor with a deceptive amount of depth and large number of features, Kdenlive could prove a little more intimidating. With a more utilitarian interface and workflow, Kdenlive is deceptive in its design, however, and generally just as easy to use as OpenShot. The KDE video editor has been updated in recent years to work a lot better on other desktop environments and is also based on the popular FFmpeg media framework, giving it a lot of compatibility with various containers, formats and codecs.

Like OpenShot, Kdenlive works on a hierarchy of separate tracks, with higher tracks having priority, and transitions being used to go between the clips up and down the tree. However, in Kdenlive there is a differentiation between video and audio tracks, allowing you to perform some audio-centric manipulation to the tracks and clips in your timeline. Effects



■ The multi-track interface is very effective

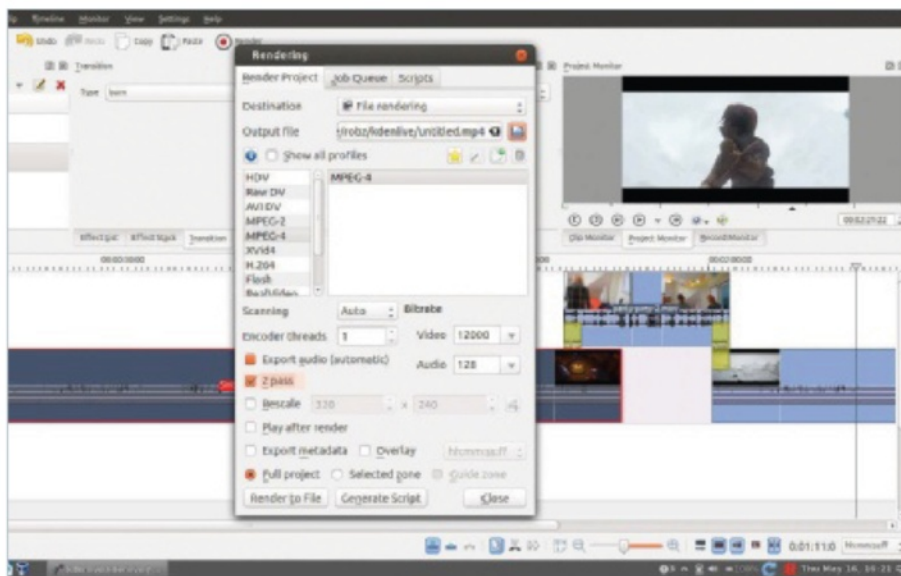
and transitions are generally easier to get to as well, with the right-click menu bringing up a list of both to access. These effects can also be heavily customised to your exact liking.

There’s a whole host of other video editing features in Kdenlive to use as well, with a stabilisation algorithm and the ability to create noise tracks, countdowns and clicks

among many others – it’s a pretty advanced piece of software. There’s also the ability to create a video DVD, taking already DVD/MPEG2 rendered material either from Kdenlive or another DVD to create your own. Rendering itself is fairly quick, with a queue so you can have multiple things ready to go at once, or even transcode other files to use elsewhere.

Kdenlive is a very complete package, with a great amount of attention to detail to a lot of its core and extended features that make sure it does just about everything you’d want it to do. With everything highly customisable, including the presets and encoding profiles, Kdenlive is a fantastic prosumer-level video editor.

“A very complete package, with a great amount of attention to detail”



■ Video exporting is highly customisable

SCORES		
Installation	Built on FFmpeg and easily obtainable	8
Ease of use	There is some sacrifice to usability for the sake of features	8
Features	A wealth of options to edit together any video project	10
Codec support	Due to the FFmpeg core, its video support extends far	9
Overall	With a little more to offer than most, and decent compromise on the interface for it, Kdenlive is one of the best Linux video editors around	9

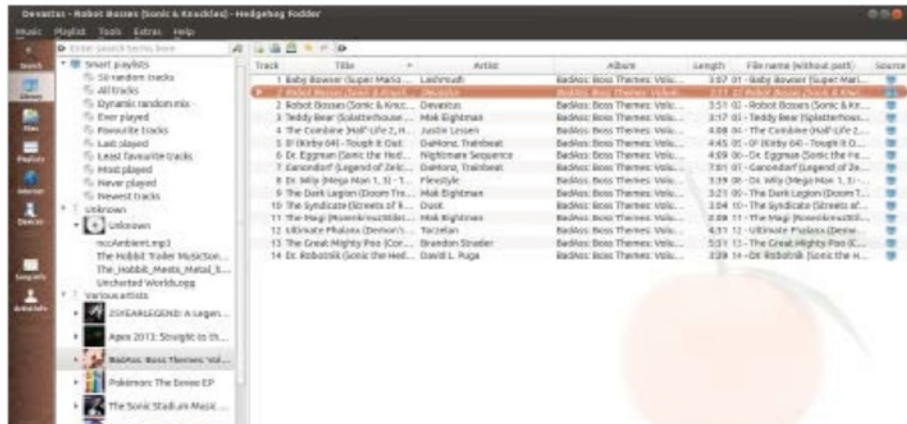
Clementine

A Linux favourite, how is the latest Clementine player?

Clementine is based on the KDE music player, AmaroK, but with a few improvements and a much better interface. It's quickly become a very popular media player and the latest version, 1.2, has arrived with a whole host of great new features. These are sure to attract new users while appeasing die-hard fans who still want to use their favourite media player in a changing landscape of music consumption.

First of all, Clementine now has access to a lot more music streaming services than before, with new additions such as Dropbox and Ubuntu One joining the already impressive list of existing ones. These include Google Drive, Spotify, SoundCloud, Last.fm and Grooveshark. You can easily search within the free services using the built-in Clementine search functions, and you can log in to do the same with the account-driven services such as Spotify and the cloud storage ones. These settings are easily found in the preferences menu under a different section to the vast wealth of customisation options that Clementine offers.

Through these options you can change just about every way Clementine behaves, from simple things like how it might fade between



■ Clementine has everything but the Kitchen Sink. It even has the Hypno Toad

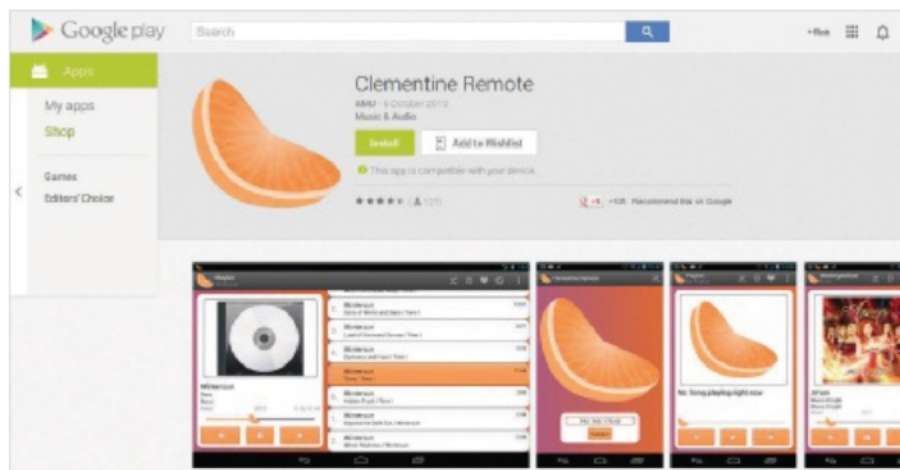
tracks, to tweaking the transcoding settings or even setting a Wii Remote as a remote control device. New in Clementine 1.2 is the ability to use an Android device as a remote, a feature which has been a long time coming. However, instead of using a basic HTTP interface, it uses a special app to make it work.

Playback is fantastic, with a special Clementine icon ticking down to the end of the song, and showing a play symbol so you know it's actually going. While you can control

Clementine from here, you can also control it from the usual volume control icons if you're using the right desktop environment.

Clementine basically has it all, then. Its smart playlist feature, the dynamic random mix, isn't quite as good as some online equivalents, but it's a lot better than any of the other players in this test. It also has the greatest selection of online services it connects to, is the most customisable and makes finding your music easy.

“You can change just about every way Clementine behaves”



■ The Android app offers better control than some HTTP interfaces

SCORES		
Playback	Makes playback as easy as it can be, short of dedicated buttons on the notification area	9
Interface	The interface contains a lot, but does the best it can for the amount of features	8
Management	Easy to navigate and find media, although some of the online services could work better	9
Online	Connects to everything you would probably want to use bar Pandora and Google Music	10
Overall	An amazing piece of software that lets you do just about anything you'd want to do with all of your music	9

Banshee

Not as popular as Clementine, but still a great option

Banshee was the one-time default audio player for Ubuntu, replacing and then being usurped by Rhythmbox. Due to this, you would be forgiven in thinking that they're incredibly similar applications – and in some regards they are. They both employ a similar three-pane layout for your media, and they both include a column down the side for navigating your media, videos, podcasts and online services. At the core, they also both run off GStreamer, which is a great media back-end and allows the two to play just about anything with the right codecs installed.

The interface for Banshee is nice and easy to use, and very responsive. Search is instant, bringing up results as you type, and the way results are listed is conducive to finding the tracks, album or artist you're looking for. The album pane on the main interface has thumbnails of the album art instead of a list – although the grid effect can be disabled if you wish. It all works very well and, like all the others, integrates just fine with the desktop



■ The Banshee interface is very nice

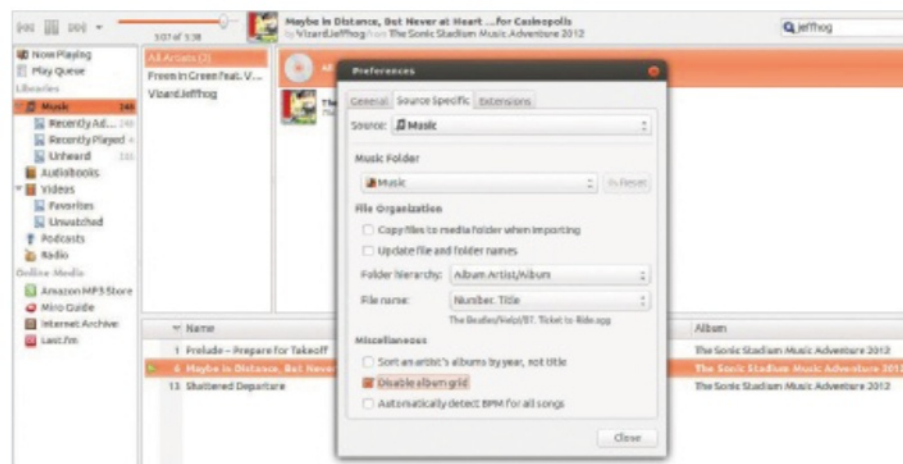
environments that allow for playback options via volume controls.

Customisation wise, there's not a whole lot more than Rhythmbox. You can't even set a specific interval or time for the music library to update. These kind of features are sorely missing, especially compared to Clementine and Audacious which have a whole host of different features and options that can help you streamline the experience. At the very least, there's a fairly rich plug-in system and you can turn off some of the features of Banshee

you don't wish this way, making it much more lightweight than it is by standard. It's through these extensions that the online services are included in Banshee – like Rhythmbox, though, there's only a handful like Last.fm and Amazon. There are a few other, community-built extensions, but none to challenge the features of Clementine.

So overall, Banshee is pretty good. While it's easy to compare it to Rhythmbox, it's generally a little better, with better plug-in support that allows it to be more lightweight if you wish, and a slightly cleaner and informative interface. It's no Clementine, though.

“Search is instant, bringing up results as you type”



■ Extensions give Banshee a lot of its features, and turning these off is the main way to customise the software

SCORES		
Playback	Good playback options, but lacks its own dedicated notification icon	7
Interface	An easy-to-use interface that is laid out in a logical manner	9
Management	For local content, it's a great way to keep track of all your media of any type	10
Online	Limited online options, and it's only minimally extendable	4
Overall	Banshee is a great media player that we'd be very happy to use if we didn't have access to any online services	8

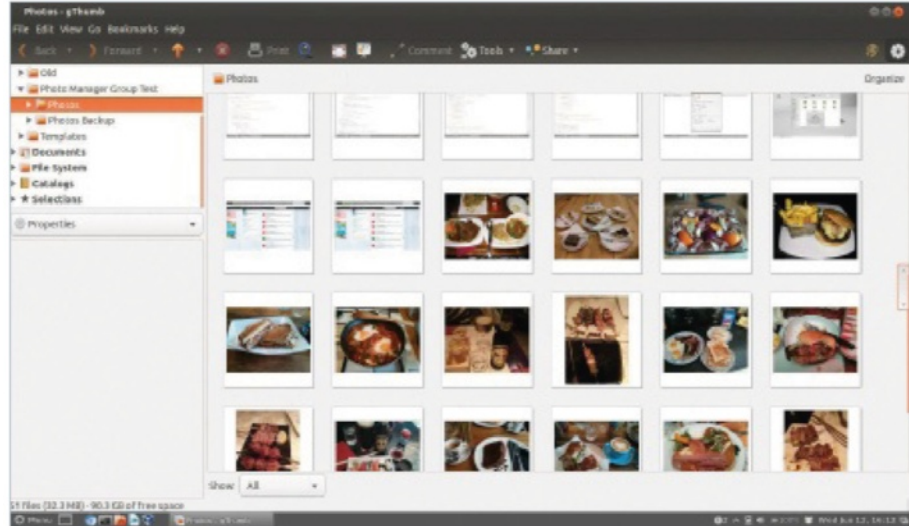
gThumb

A feature-packed photo manager complete with editing options

We were actually a little surprised by gThumb, especially as it's a GNOME-made application. With the recent move to simplifying all parts of GNOME, we were wondering whether its popularity was misplaced; however, it looks like all the changes to the GNOME Shell haven't affected the way gThumb works.

It's not to say it's a super-advanced, though – it features a fairly straightforward interface that we'd want from a photo manager. By default, it's a thumbnail view, displaying the Picture folder in the home directory. Instead of importing photos from specific albums, you merely navigate through the directory structure. This means that instead of having specific albums to create, you'll need to organise those photos yourself in a file manager or terminal.

Clicking on the Edit file button in the top right opens up the image editor, and a sort of limited filmstrip view as well. You can click between photos in the stream on the bottom to edit them individually, although you can't move between them with arrow keys. The editor is



■ gThumb lets you navigate the directory structure

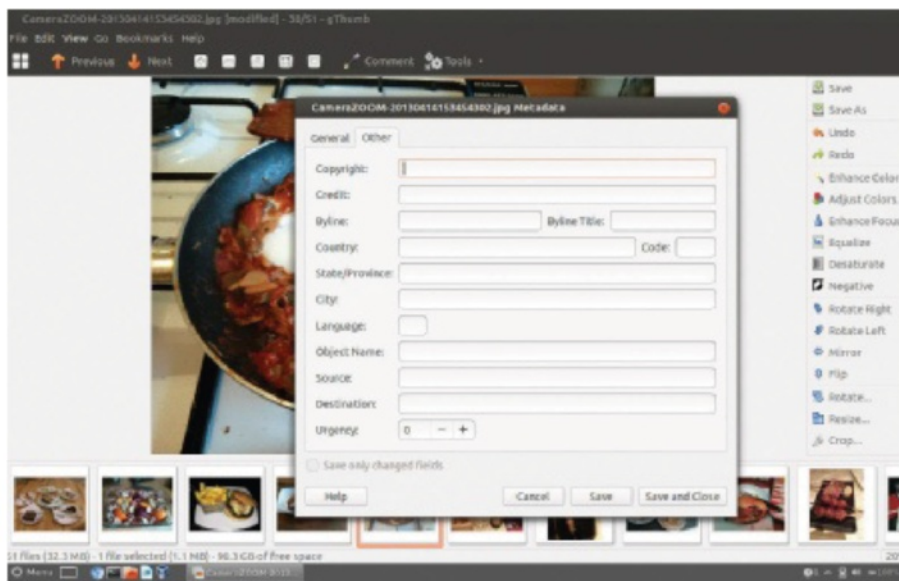
a little more advanced than what you get with Shotwell, allowing for more manual colour and balance corrections, as well as anti-blurring, desaturation and negative filters. There isn't any red-eye removal, though, and no paintbrushes or anything to do that, meaning you'll need to open GIMP for those kind of operations.

You also can't do batch autocorrection for the images, but you can do batch tagging. You can drag a box over the images to select them, or press Ctrl/Shift with click, and you'll be able to add, remove or assign tags in batches. You can edit information in batches as well; however,

you'll likely want to do that individually. Images that have been edited in this way get a date added to them, letting you know when they were last modified by gThumb.

Like Shotwell, gThumb also has online publishing tools, with similar services such as Facebook, Twitter, Flickr etc. It's nice that these tools exist, as these services don't have touch-up options, allowing for mass uploading of holiday photos and the like.

gThumb is a great little application and while it doesn't quite have everything we want, it's definitely a step up from Shotwell.



■ Plenty of photo information can be changed in gThumb

SCORES		
Installation	Easy to obtain, but also requires a few GNOME packages	8
Ease of use	Generally very easy to use; however, the edit/filmstrip view is confusing	8
Editing	Decent amount of editing tools, although no red-eye removal	7
Features	While not particularly feature-heavy, it's got enough to properly manage your images	7
Overall	A very strong showing from gThumb, a photo manager with plenty of features, although it could do with some batch image touch-up options	8

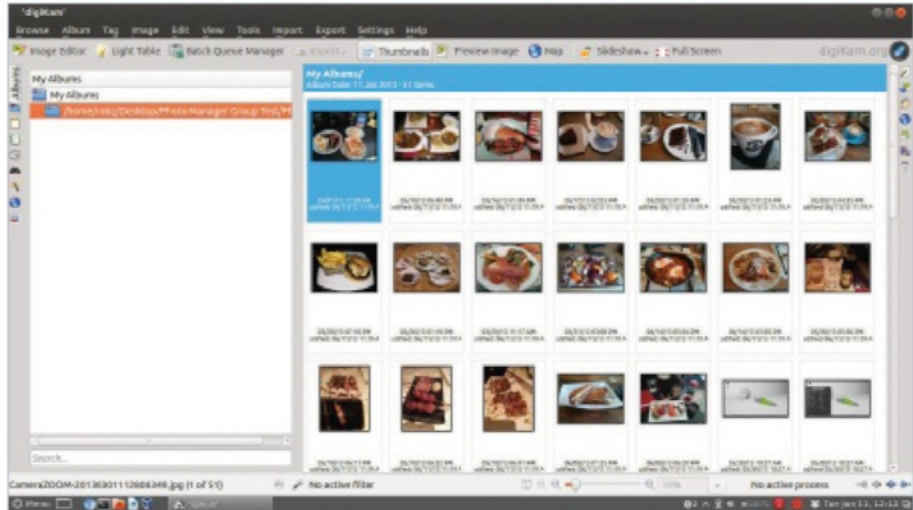
digiKam

A fantastic and comprehensive photo managing application

The KDE-developed image organiser is now in version 3.x, receiving a few upgrades over the past two years since it hit version 2.0 and became one of open source's premier photo managers. It comes with a full range of functions, including the standard preview and filmstrip view we're looking for, batch tag and data editing, editing software, and even a light-table function.

Although built for KDE, digiKam works just fine with GTK-based desktop environments and has a more neutral design aesthetic than KDE, allowing it to not stick out on other desktops. It can also be found in all major package managers thanks to its popularity and ties to KDE, although options to compile from source via a tarball or GitHub are just as available.

Initial startup involves a quick but thorough setup wizard, going through basics such as photo library locations, the manner of importing RAW files and a few more options. While a lot of people can mostly ignore these and have a decent default selection, they are useful for people with more specific needs. Any changes you make can be altered in the main program's settings and such, and there's a lot of extra customisation you can do to digiKam as well.



■ digiKam has a selection of different views

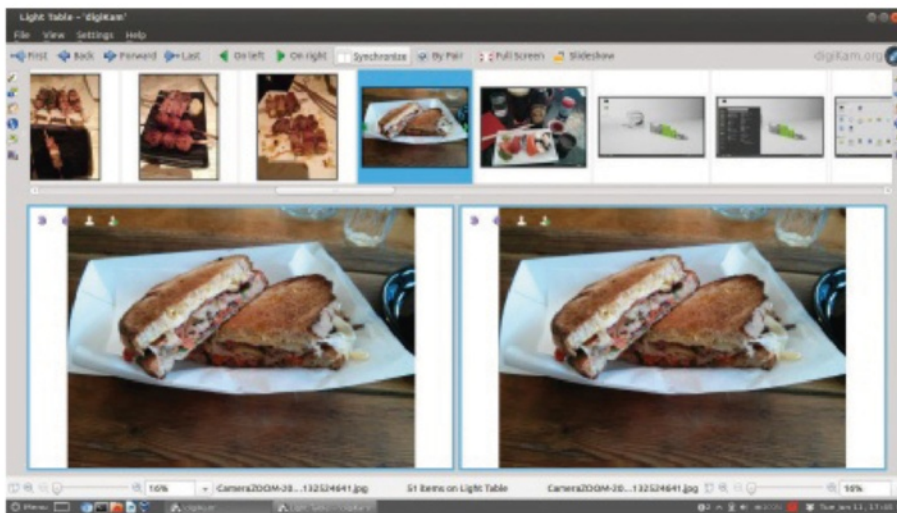
The main window allows you to view your photo library in a stream, with thumbnails and previews, as well as by their geotags if they have any. It's here you can edit tags, geotags etc, as well as looking up metadata and colour maps for the images. The interface aids in this, allowing you to select multiple images with the plus symbol that appears as you hover over thumbnails, although Ctrl and click also still works.

Selecting an image then allows you to enter the image-editor window or the light-table view. In the image editor, there are a lot of functions for adding effects and general image manipulation, although it's mostly hidden away in drop-down menus, making it a bit tricky to use. The light table is a great feature, allowing

you to compare and contrast a couple of photos in case you need to select from multiple images of the same angle and subject.

It's a really comprehensive piece of software, offering just about everything we'd want from a photo manager and more, especially for those that need it in a professional capacity.

“It has everything we'd want from a photo manager”

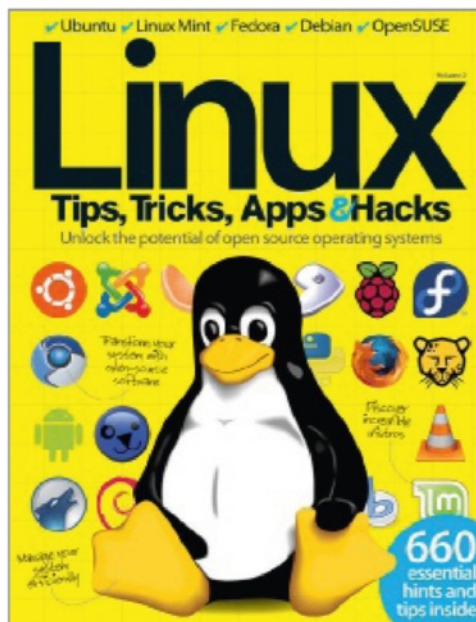


■ Light table allows you to compare photos to find which is best

SCORES		
Installation	Requires some extra packages from KDE, but easy to install otherwise	8
Ease of use	The main window is easy enough to use, although the image editor is a little confusing	7
Editing	Not quite at the level of GIMP, but more than good enough for a photo manager	9
Features	digiKam is overflowing with a variety of features to make it easier and better to use	9
Overall	A fantastic overall package that offers specialised tools to organise, analyse and edit your photo library	9

Special trial offer

Enjoyed this book?



Exclusive offer for new



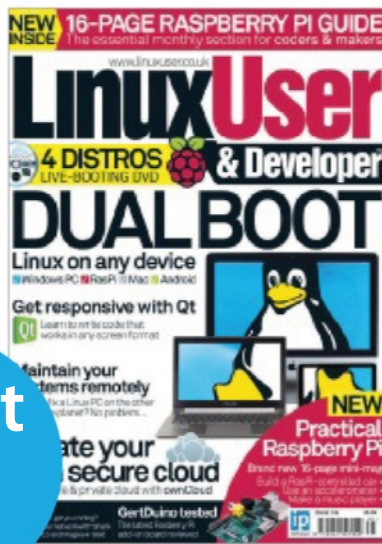
Try 3 issues for just £5*

*This offer entitles new UK direct debit subscribers to receive their first three issues for £5. After these issues, subscribers will then pay £25.15 every six issues. Subscribers can cancel this subscription at any time. New subscriptions will start from the next available issue. Offer code ZGGZIN must be quoted to receive this special subscriptions price. Direct debit guarantee available on request.

** This is an US subscription offer. The USA issue rate is based on an annual subscription price of £65 for 13 issues which is equivalent to \$102 at the time of writing compared with the newsstand price of \$16.99 for 13 issues being \$220.87. Your subscription will start from the next available issue.



About
the
mag



**Dedicated to
all things Linux**

Written for you

Linux User & Developer is the only magazine dedicated to advanced users, developers & IT professionals

In-depth guides & features

Written by grass-roots developers and industry experts

4.5GB DVD every issue

Four of the hottest distros feature every month – insert the disc, reboot your PC and test them all!

subscribers to...

LINUX USER & Developer™

Try 3 issues for **£5 in the UK***
or just **\$7.85 per issue in the USA****
(saving 54% off the newsstand price)

For amazing offers please visit

www.imaginesubs.co.uk/lud

Quote code **ZGGZIN**

Or telephone UK 0844 249 0282 overseas +44 (0) 1795 418 661

We don't keep secrets



Tips & Tricks

Learn the truth about iPhone, iPad, Android, Photoshop and more with the Tips & Tricks series' expert advice and tutorials

BUY YOUR COPY TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



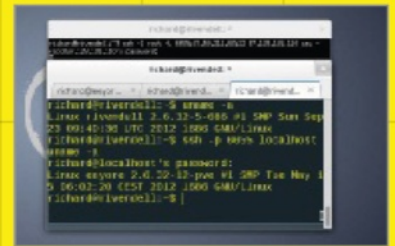
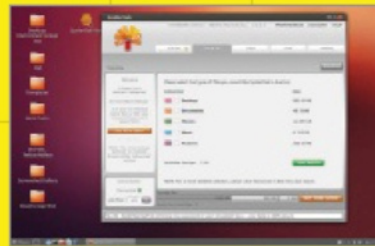
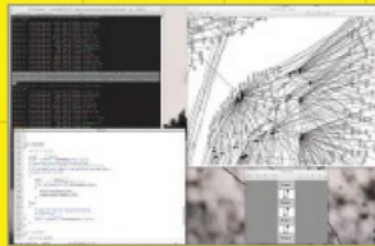
 facebook.com/ImagineBookazines

 twitter.com/Books_Imagine

✓ Ubuntu ✓ Linux Mint ✓ Fedora ✓ Debian ✓ OpenSUSE

Everything you need to get the most from Linux

Top hints and tips to guide you through the best open source software and operating systems



Tips

Program with Python and protect your network with step-by-step guides

Tricks

Maximise the potential of open-source software with comprehensive tutorials

Apps

Discover top apps and distros that will enhance your Linux-based projects

Hacks

Improve your Linux experience by tweaking & customising your system



Tutorials and guides on all major distros

Protect and secure your PC

Turn your Raspberry Pi into a router

Master dual booting with Windows

ip **DigitalEdition**
 IMAGINE PUBLISHING GreatDigitalMags.com
 VOLUME 02

www.imaginebookshop.co.uk

In-depth guides and tutorials written by Linux experts