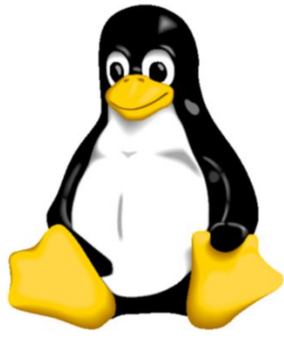
Linux & Open Source Genius Guide

The essential guide to mastering open source software and operating systems

Over 200 pages of expert tutorials and features!



Linux & Open Source

Genius Guide

Genius Guide

Linux & Open Source

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

Editor in Chief
Dave Harfield

Production Editor
Russel Barnes

Design
Danielle Dixon

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.

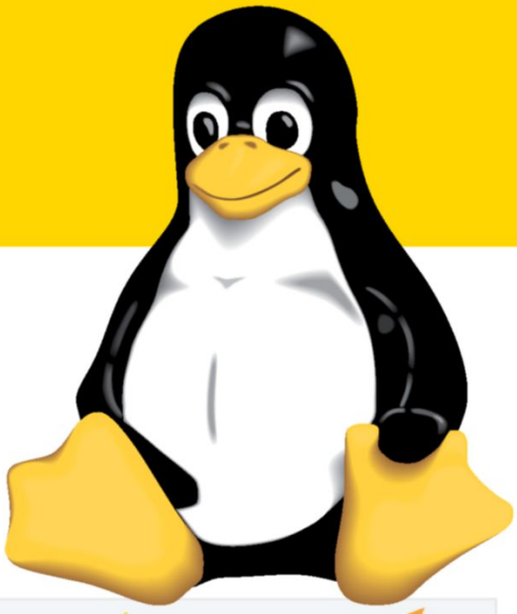
ISBN 978 1 908222 2 37

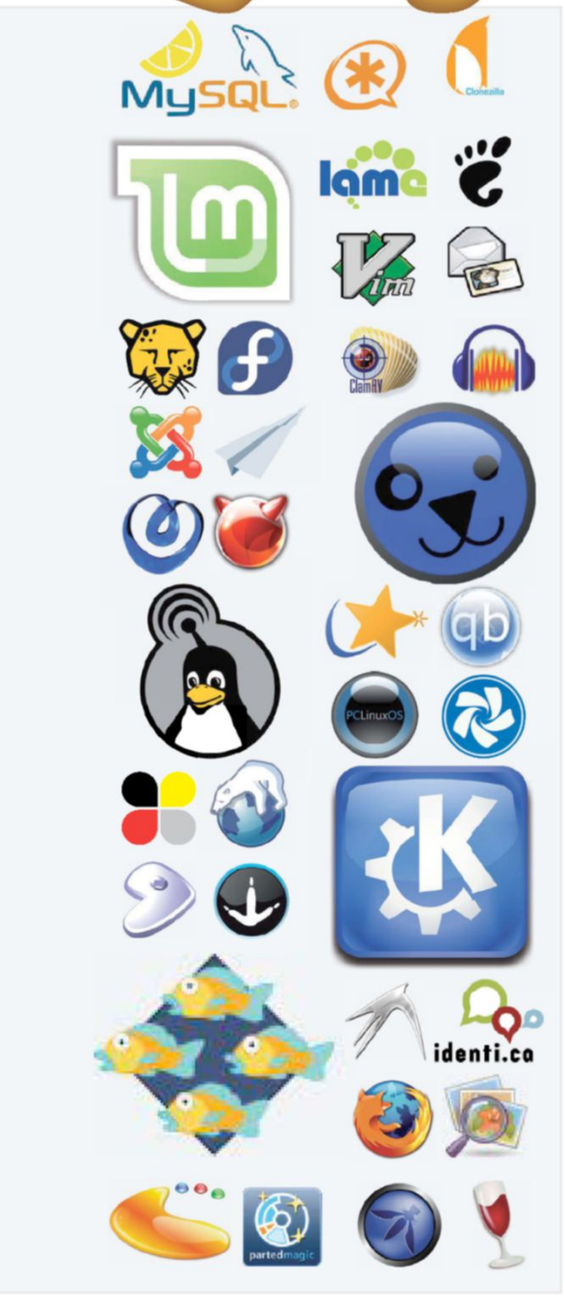
Contents

Your complete guide to the Linux and open source scene

TIPS & TRICKS

2 minute tutorials

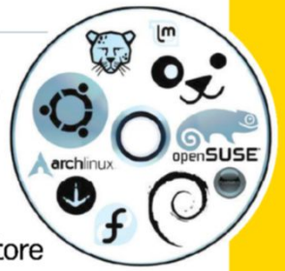




Linux Masterclass

Developer guides

Essential software



On the disc

We asked our readers. We've asked open source luminaries. We've asked our writers. And we've been pooling our own views too. It's with great pleasure that we celebrate Linux, therefore, with 100 open source projects chosen by the Linux User & Developer community...

There's a reason that very few people attempt to pull together a feature such as this. The open source community has, over an extensive period, offered an unprecedented level of software choice. Every year, thousands of new projects are worked upon, some successfully, some not. But the vast majority work on the ethos of sharing, and of giving something back.

There seemed little point, then, even attempting to distil those hundreds of thousands of projects and programs down to a list of the best 100, because one person's chalk is another's cheese. However, what seemed more interesting was to sound out some of the big names in the open source community. To take the views of our writers. To find out what you, the readers, think. In short, to take a community approach to a community feature. Warts and all.

And it's from a combination of those three groups that we present 100 of our favourite open source projects and programs, which celebrate the breadth, talent and downright quality of the open source community. There's a lot to celebrate, but a lot we've left out. In fact, it's inevitable we'll have omitted one of your favourites. And in the spirit of the community, this is a feature that should evolve, and should grow. So do take the time to head over to our forum at **www.linuxuser.co.uk** and add your own recommendations, to share with other readers.

But for now, here are 100 open source programs and projects that might just get you started. Let the party begin...

EMAIL & COMMUNICATION

MOZILLA THUNDERBIRD

Web: www.mozillamessaging.com
Get Involved: developer.mozilla.org/en/Thunderbird

Inevitably, it's Mozilla that's once again made a sizeable splash here, building on the strength of the Firefox web browser and matching it with a project that manages email and news.

Formally released at the end of 2004, Thunderbird unsurprisingly works on the same ethos as Firefox. Thus, the central core program is easily scalable via community plug-ins, and it's up to you how much you choose to expand the central proposition.

Even if you don't, Thunderbird, while not without its critics, is a strong piece of software. Its security and low system overhead are two key advantages, certainly (it's understood that the French government has now adopted it, giving some sign of how trusted and comparably safe it's become). The ease-of-use and delivering far more than you'd expect for free are the icing on the cake. It may not be quite as high-profile as Firefox, but Thunderbird is an excellent email client, and deserves equal levels of exposure.

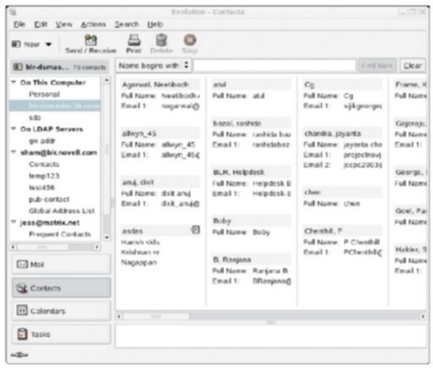
PIDGIN

Web: www.pidgin.im
Get involved: developer.pidgin.im

In the olden days, many of us were chatting to each other over ICQ. Then, big companies decided they fancied a slice of the pie, and along came MSN Messenger, Yahoo! Messenger, AIM and many more. In fact, they keep on coming, with Facebook chat the biggest threat to many of the traditional instant messengers of old.

Pidgin does its best to be all things to all people, and achieves far better results than it has any real right to. It's a universal chat client that works across the main instant messenger services. Even Facebook support can be added via a plug-in, should you so desire.

Crucially, though, Pidgin just works. You don't always get the precise, individual features that each IM tool likes to offer you, but we can happily live without being diverted to an MSN page every time we log into something like MSN Messenger. No, what Pidgin does is not take its eyes off the proverbial ball. We use it because we want to chat to other people, regardless of service or platform. And that's just what Pidgin achieves.



Also consider:

EVOLUTION projects.gnome.org/evolution
The popular Evolution is an email, calendar and address book client aimed at those who use the GNOME desktop. The feature list for the project is extensive, and it's a popular alternative to Thunderbird.

SENDMAIL
www.sendmail.com/sm/open_source
Sendmail continues to exist in open source form. It supports lots of different ways of transferring email from A to B, and most of the world's email is routed by this humble piece of FOSS code. It carries the endorsement of Jon 'maddog' Hall, who highlights it "not only because it carried a lot of email, but it was first/best in class."

LIGHTNING
www.mozilla.org/projects/calendar/lightning
If you're looking to stay with Mozilla, and want just the calendar functionality that's present in most email applications, then Lightning has become the heir apparent to the likes of Sunbird. Compared to many of the projects here, it's in its infancy, but is one to watch.

SEAMONKEY www.seamonkey-project.org
Picking up the reins of the Mozilla Application Suite of old, what SeaMonkey proves is that you can have a strong collection of internet tools gathered in one place, under one umbrella. It is flexible in its licensing and more community-driven than Mozilla's projects now appear to be. It's also the kind of download that's likely to offer something you like, even if you don't need the whole suite.

ASTERISK www.asterisk.org
Asterisk is a piece of software that can turn your computer into a communications server. It's powerful stuff, easily scaled to organisations of pretty much any size, and it can handle VoIP, conference servers, PBX and a lot more. As Jon 'maddog' Hall says: "a great PBX at an affordable price. I have to state that as a former 'telephone company guy' I was very impressed about this project from the start. Not only did it foster a 'support company' (Digium), but thousands of smaller VARs around the world that did installation and support off Asterisk. I am particularly proud because Mark Spencer, the founder and architect of Asterisk, credited one of my talks in 1999 with convincing him to make the licence 'open source'."

DING www-user.tu-chemnitz.de/~fri/ding
Finally, if you're communicating across language barriers, then a recommendation from Henne Vogelsang at SUSE: "I'm not a native English speaker and I manage a lot of moving parts in my FOSS live. I constantly learn things and I constantly forget things. That includes a nice chunk of the English vocabulary, unfortunately. In Ding I can look up words, use them and then forget them again. They never even grace my long-term memory. The dictionary of Ding is just great; it includes phrases, foul language and other trivia and it's local so I don't have to rely on the network to be able to express myself freely. The interface, although old and in TCL/TK, is simple, clean and very fast. I don't know anything that compares even remotely to it."

GRAPHICS

GIMP

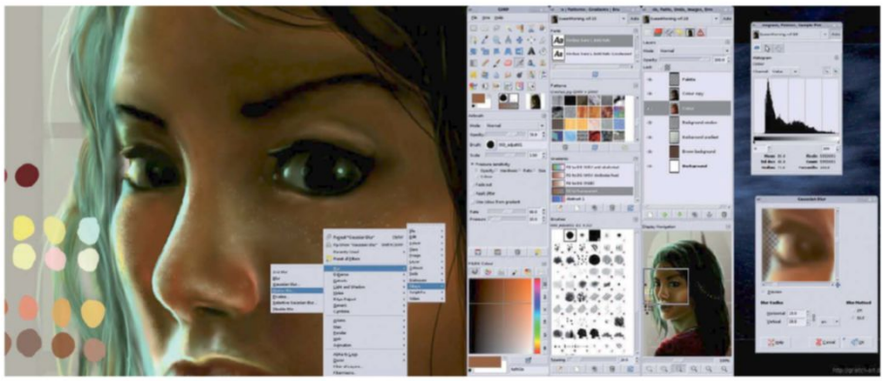
Website: www.gimp.org

Get Involved: www.gimp.org/develop

"I saw GIMP, and I knew that open source would win." The words of Jon 'maddog' Hall, and he perfectly captures just what GIMP (which stands for GNU Image Manipulation Program) meant when he says "it was at the Red Hat 'offices' in the early days of their company, and to see GIMP manipulating graphical images along the lines of Photoshop, and not having to pay zillions of dollars for it... I was impressed."

Subsequently, many millions of others have been impressed, too. Because one substantive factor that GIMP addressed was that people had been equating free software with products that were somehow lesser than, and not as powerful as, their commercial cousins. GIMP, a raster graphics editor at heart, has gone toe-to-toe with the kind of product that one company can spend thousands of pounds on licences for, and made real headway.

Furthermore, it's done it with different ideas. The interface for GIMP, and its working methodology, is no thinly veiled clone of Photoshop. Instead, it's genuinely bold, risky and inevitably leaves some cold as a result. For those who do warm to GIMP, though, it becomes the kind of application that's simply tricky to live without.



Also consider:

DIGIKAM www.digikam.org

"Aimed at amateurs and serious photographers alike, digiKam boasts functionality that covers the entire photographic workflow: from importing and organising photos to batch processing and sharing them," says Dmitri Popov. "Geotagging, support for filtering, and advanced search capabilities are just a few of the features that this powerful photo management application puts at your disposal. Professional photographers will appreciate that digiKam can handle RAW files from virtually any camera on the market. The application also sports advanced features like lens distortion correction, colour management and a slew of advanced editing tools."

INKSCAPE www.inkscape.org

There's no shortage of quality graphics tools and applications in the open source community, and this is very much an area it specialises in. Take SVG graphics editor Inkscape, which takes on the not-inconsiderable commercial might of Adobe Illustrator and Corel Draw. With genuine success, too.

QCAD www.ribbonsoft.com/qcad

Computer aided design (CAD) is the kind of software, too, in the proprietary world that can leave a sizeable hole in your pocket. Yet projects such as QCAD are amongst those leading the open source alternatives. Granted, there's a professional edition for which you pay, but a community edition has also been released under the GPL.

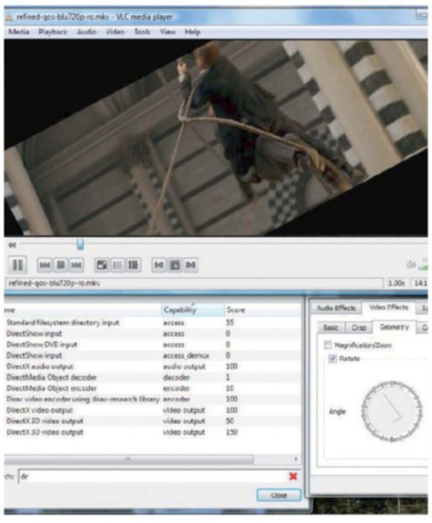
GTHUMB live.gnome.org/gthumb

A useful tool for those with a GNOME desktop is gThumb. This is an image browser and viewer, one that can also grab images directly from digital cameras if required. It's small, but useful.

SHOTWELL yorba.org/shotwell

If it's specific photo management you're looking for, Shotwell can import, organise, let you edit and then publish your shots. You'll find it included by default in a couple of popular Linux distros, and for good reason. Its comfortable support of the RAW format, for starters, is a good reason for its appeal.

MEDIA PLAYBACK



VLC PLAYER

Website: www.videolan.org/vlc

Donation page: www.videolan.org/contribute.html

Switch on a Windows PC and try something like Cyberlink's PowerDVD and you'll find yourself deluged with update requests and registration forms. But VLC Player? Someone remembered the golden rule of good software: it never gets in the way of what you want to do.

Managing a vast breadth of codecs, and allowing the easy and flexible playback of a whole host of media, it's a low footprint application that does its work and barely lets you know it's there.

Dominique Leuenberger (aka DimStar), one of the core GNOME maintainers, captures the strength of VLC perfectly. "This was probably one of the first open source projects I had contact with, without really knowing about its real roots," he told us. "But having looked for a decent media player, I came across that one. And boy: calling it a media player should probably be rated as an insult by now. It is so much more than that, including a full-blown VOD server". He's right, too.

Also consider:

AMAROK amarok.kde.org

Amarok's breadth of support for the likes of Last.fm, UPnP, MusicBrainz and more helps lift it above the usual, run-of-the-mill music players. Amarok, to be fair, is already mature and well above most of its rivals, and is well worth tinkering with.

BANSHEE banshee.fm

Who needs the likes of iTunes? Banshee does a terrific job of bringing together audio and video playback, as well as allowing you to sync mobile devices and subscribe to podcasts. All without Apple DRM!

OFFICE

LIBREOFFICE

Website: www.libreoffice.org/download
Get involved: www.libreoffice.org/get-involved/

It's a sign of the ethos of the open source community that, even a year ago, we'd have been talking about OpenOffice here. However, politics have dictated that there's been a fork in the road and while OpenOffice stays under the ownership of Oracle (infamous, of course, for closing down OpenSolaris), LibreOffice is the splinter that's under the stewardship of The Document Foundation.

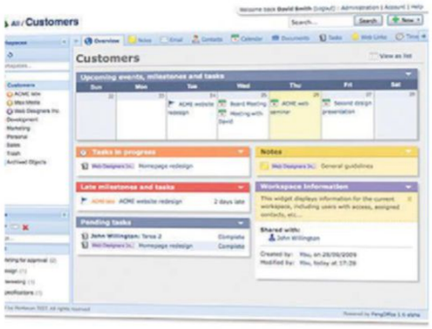
It's an immense piece of work, too, that has significantly closed the gap on Microsoft Office in terms of features and users, aided by how comfortably cross-platform it is. What makes it easily one of the most significant inclusions in this feature is how it has brought open source to the attention of many. End-users, fed up with paying three figures for decreasingly improving proprietary office solutions, find it ample, while the per-seat saving for corporates is even more notable.

Factor in the broad file support and the breadth of applications within LibreOffice and it bears comparison with any of its peers, free or paid-for. As Adrian Bridgwater says, it's "done a great job of keeping the community spirit alive, and even been showcasing its collective work at a few high-profile IT shows by all accounts."

SCRIBUS

Web: www.scribus.net/canvas/Scribus
Get involved: www.scribus.net/canvas/Join

We suppose that this just about sits under the office banner, although it also has a strong part to play in creative work. But then Scribus is an adaptable beast, the premier open source desktop publishing/page layout package, and the one tasked with going head to head with the likes of Quark XPress and the market-dominant Adobe InDesign.



It doesn't skimp on professional features as a result, and it's been proven on more than one occasion that it's a tool that can indeed produce quality end results. Crucially, they're end results that can stand toe-to-toe with the output of the aforementioned Adobe and Quark packages (although, of course, the talent of the user is generally the defining feature).

Scribus is far from a perfect beast, but its rate of development continues to address this quickly. This is one of the more active open source projects out there, and one that's constantly seeking further help and support. Do consider giving it if you can.

Also consider:

CALLIGRA www.calligra-suite.org/get-calligra
Open source has no shortage of quality office suites. Try the Calligra Suite, for instance, which is also cross-platform and supports the OpenDocument file format (and a fork of KOffice). It's KDE-dependent and generally brings with it a lower footprint than LibreOffice.

FENG OFFICE www.fengoffice.com
The future, meanwhile, seems to be cloud-based office tools such as Google Docs. Google Docs is closed source, however, as is Zoho Office. For open source alternatives, the future may well lie in the hands of something like the quietly impressive Feng Office Community Edition, which is well positioned to take up the mantle and adds strong project management abilities to the mix.

GNUCASH www.gnucash.org
Next, GnuCash, which is just the kind of tool that can greatly benefit a fledgling business. It's a piece of financial accounting software which easily lets you keep on top of your cashflow, and it takes no shortcuts in the way it goes about its business. One of GnuCash's great strengths is just how often it is improved and updated, and it's backed by a sizeable community, too.

FOCUSWRITER gottcode.org/focuswriter
Meanwhile, if it's a very concentrated text tool you're after, then Jos Poortvliet, community manager at openSUSE, recommends FocusWriter. "If you have to write a text," he told us, "FocusWriter takes away all distraction, being a full-screen interface showing text and nothing else. It is easy to use and helps me increase productivity."

PDFEDIT pdfedit.petricek.net
As its name implies, PDFEdit lets you edit PDF files. Crucially, it does it without clogging up your system *à la* Adobe Acrobat. The functionality of the software is scalable, but the guts of it alone will satiate many.

"I saw GIMP, and I knew that open source would win"
Jon 'maddog' Hall

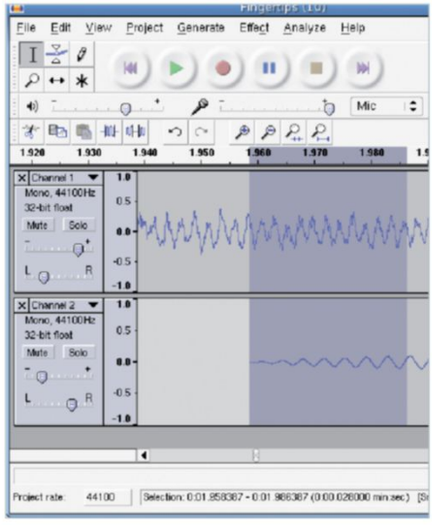
AUDIO EDITING

AUDACITY

Website: audacity.sourceforge.net/
Get Involved: audacity.sourceforge.net/community/

In 2009, the film *Paranormal Activity* became one of Hollywood's biggest surprise hits of all time. It was filmed by a man called Oren Peli, who shot the movie in his home and edited it together himself. And as you can see if you look at the film closely, he used Audacity to do much of the audio work on the film. The same film that went on to make $193m at the box office.

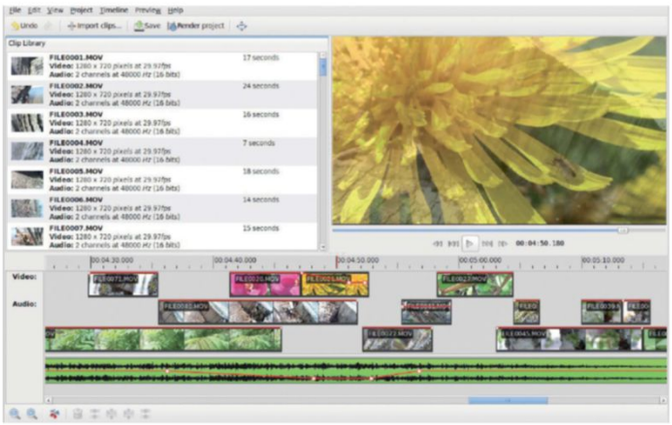
Audacity is just the kind of quiet, powerful tool you can call on in a crisis, though. It's fussy about some formats that it won't support (and you'll need to hunt down an MP3 encoder, should you wish to use one), but its strengths are numerous. It's easy to use, offers sizeable power over what you can do with your audio, and generates results that – as the aforementioned *Paranormal Activity* proves – can work commercially. It's worth, especially, keeping an eye on the regular beta versions, too, to keep on top of Audacity's latest ideas.

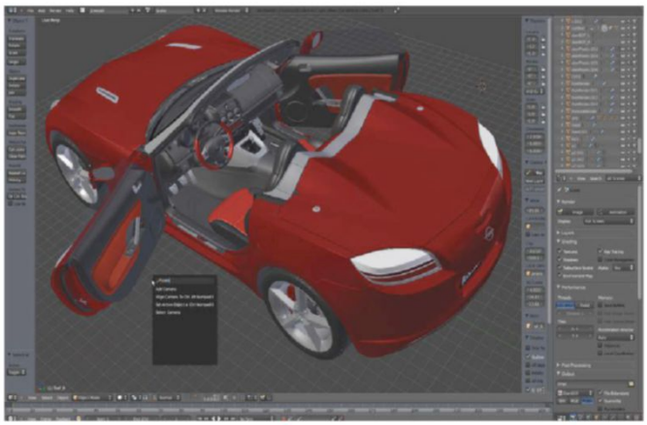


INTRODUCTION

Jim says

"This one might seem obvious but it's easy for me: the kernel. While LUD celebrates its 100th issue, we're preparing to celebrate the 20th anniversary of Linux this year. Linus certainly didn't intend it when he first shared Linux, but the OS has inspired hundreds of today's open source software projects and companies. It is the largest collaborative development project in the history of computing and it's everywhere: planes, trains and automobiles; phones, homes, TVs, ATMs...





MEDIA CREATION & OUTPUT

BLENDER

Web: www.blender.org

Get involved: www.blender.org/development/

Blender can model 3D work. It can animate it. It can composite, render and allow you to create quality interactive 3D work. The end results can be stunning. It's the kind of software that everyday users wouldn't get close to in days of old, requiring them to go to a place that could afford a licence just to get their hands on it.

Inevitably, any discussion of Blender involves a degree of gushing, but it's justified. There's good reason why books and DVDs have been produced about Blender, because it's massively sophisticated and capable of generating professional-quality content (as it, indeed, does). Want examples? Well, it was used in *Spider-Man 2*, for starters, and has since been used in a number of TV and film productions.

And you can try it for free. More than that, you can use it for free. Make the most of it for free. And generate the kind of work that you may not have realised you were capable of, all for free. In short, it's one of the biggest gifts open source software has given to the creative industries. No question.

Also consider

DEVEDE www.rastersoft.com/programas/devede.html

Authoring DVDs can be a tricky and fiddly job, but one that tools such as DeVeDe have addressed. DeVeDe allows you to easily create video CDs and DVDs, and it can output content to disc, portable media players or online. Its broad file and codec support is a real advantage, too.

BRASERO www.gnome.org/projects/brasero

For straightforward disc burning, Brasero comes warmly recommended. Designed for the GNOME desktop environment, simplicity is very much the key here. It basically offers a graphical front-end for a handful of existing tools, and with minimal system footprint.

OPENSHOT www.openshot.org

If you're looking to edit video into shape before you output it, however, then you might like to try something like OpenShot. It's a popular alternative to proprietary tools such as Adobe Premiere Elements and Pinnacle Studio for Windows, and it adopts a similar way of working to each of them. The traditional timeline is present and correct, and codec support is staggering.

PITIVI www.pitivi.org

You could also give PiTiVi a run. It boasts that it can handle pretty much any format that you throw at it, but just as significant is the fact that it attempts to scale itself to the expertise of whoever happens to be using the software at the time. It's an impressive piece of work.

ARISTA TRANSCODER launchpad.net/arista

For making transcoding work look simple, Arista Transcoder is a small download – one that allows you to rip from the likes of DVDs and to easily convert media content into the appropriate format for the device you're targeting.

HANDBRAKE handbrake.fr

Of course, you could always opt for the better-known HandBrake, and you're unlikely to be disappointed with that, either. You get lots of regular updates as well as broad, platform-agnostic transcoding functionality.

LAME lame.sourceforge.net

We should also give a present and correct salute to LAME, the popular MP3 encoder. Available as a plug-in for lots of other tools, in order to get MP3 support up and running (Audacity, for instance), it's an acclaimed encoder, and a widely used one.

OGG VORBIS www.vorbis.com

Even better, but lesser used, is Ogg Vorbis. This is patent-free, entirely open and increasingly used. But it's got some way to go to dislodge the dominant MP3 format for digital music.

MEDIA CENTRE

XMBC xmbc.org

There's a lovely irony that the XBMC Media Center actually rose to prominence because of a Microsoft product. The product in question was the original Xbox console, and XBMC is a media-centre project that ran on the machine. But, fuelled by the open source movement, it's become a lot more than that. Working across assorted operating systems and platforms, it's a popular and wise choice.

FREEVO freevo.sourceforge.net

Freevo is a rounded media-centre suite that brings together digital video recording and PVR features with management of your media and the capability to play it back using existing open source software – which it wisely incorporates. Its rate of update isn't too quick, sadly, but it's hard to quibble with the end results.

TOOLS

There are plenty of handy little tools that the open source community has generated, and these alone could fill the pages of this magazine many times over. Instead, here are a few of your favourites.

"Packaging and distributing software is and has always been a challenge in the world of free software," says Jos Poortvliet of openSUSE, "and I'm really happy that there are three projects now which together can solve this issue…"

OBS
"OBS provides a build technology which can generate packages for basically any Linux distribution, if needed, and even other OSs like Windows (used by VLC) and Mac. It can even be used to build entire distros."

APPSTREAM distributions.freedesktop.org/wiki/AppStream
Next of the trio? "AppStream is a collaborative effort to bring the Linux distributions around a table and find as much common ground for application installation as possible. Sharing screenshots, ratings, comments and possibly standardising on UIs."

"Packaging and distributing software is and has always been a challenge in the world of free software"

Jos Poortvliet of openSUSE

Bretzn dramatically simplifies the deployment of software for Linux

BRETZN www.socialdesktop.org/bretzn
"Bretzn," meanwhile, "brings OBS, AppStream and other technologies together to dramatically simplify the deployment of software for Linux, automatically building and distributing software from within an IDE plug-in".

HADOOP hadoop.apache.org
Adil Mezghouti gives Hadoop a shout. The project, which is based around servicing reliable distributed computing, earns its recommendation "because of the power it gives users who need to process a lot of data."

BEAGLE www.beagle-project.org
For desktop searches, Beagle is excellent. Development appears to be on hold, which is a shame, but the last release is nonetheless a terrific way to hunt down and index a real breadth of material on your computer.

OWASP www.owasp.org
LUD reader Simon Whitehouse recommends OWASP. "It's an initiative to help companies and individuals improve application security," he argues, "which many business leaders and developers do not factor in when they design, make and implement their systems."

PING
openSUSE's Vincent Untz is a passionate advocate of Ping. "Ping is one of those very little utilities that is taken for granted by many, and that is used by even more people. What I really love about it is that it is so simple (both implementation-wise and from a user perspective) and yet, it is extremely useful."

MEMTEST www.memtest.org
When you've just built or deployed a new machine, it's worth doing some soak-testing work to see if it can handle being pushed particularly hard. That's where a little open source tool such as Memtest will come in. The basic idea is that the program stresses the machine in question, hunting down errors that occur when the machine is under load. It runs a variety of tests and can identity a hardware problem before it's had the chance to become one.

VIRTUALISATION

It's testament to the power of modern-day computing that virtualisation, in some cases, can appear effortless. However, there's hard-working software involved, too, and the open source community has no shortage of it.

VIRTUALBOX www.virtualbox.org/wiki/VirtualBox
VirtualBox, for starters, is aimed at servers, desktops and embedded platforms, and is able to virtualise x86 hardware. It means, of course, that you can run operating systems effectively within operating systems, but there's plenty more to it than that. Easily one of the most popular applications of its ilk, in both the proprietary and open source worlds, it's a sophisticated, complex piece of software that makes its job look easy.

WINE www.winehq.org
Wine, for those looking to run Windows applications on their machines, is quite brilliant, too. Capable of running a massive number of Windows applications, including some taxing games, it's been a real help for those looking to migrate from Windows to Linux, and for those who have to work across both platforms.

ODDS AND ENDS

FREEMIND freemind.sourceforge.net
Some get nothing whatsoever out of mind-mapping software, others find it invaluable. If you're in the latter camp, then FreeMind is a quality open source project that may prove of real value. It depends, of course, if your brain works in the manner in which the software wants it to. But success rates are high and it costs nothing to try.

CELTX www.celtx.com
Specialist screenwriting applications usually cost hundreds of pounds to buy, and given the strict formats that professional screenwriters need to adhere to, it's money that in the past had to be spent. That was, at least, until Celtx came along. It's a project that's still a little rough around the edges, yet it's also one with lots up its sleeve. Screenwriting is one facet of the project (with plenty of useful templates), but pre-production planning and project development can also be handled via it.

SOCIAL & BLOGGING

WORDPRESS

Website: www.wordpress.org
Get involved: www.wordpress.org/about

If you're looking for an example of how open source has levelled the playing field for a mass audience, then see how long it takes a novice user to get a website up and running from scratch, purely using WordPress and nothing else. Then, go a stage further. Dig around at the bottom of some pretty advanced, and very popular, sites and scroll right down to the bottom. Don't be surprised if there's a WordPress credit down there. What started as a tool for blogging has grown into something extremely significant.

It's open source at its best (having started as a fork of the b2/cafelog project), offering umpteen ways for people to utilise the product, and also to contribute to it. The web is awash with WordPress templates, ideas and plug-ins, and its growth continues to impress.

LUD reader Craig Jones notes that "as a PHP developer, I'm not usually a fan of off-the-shelf packages. I like to build bespoke websites and applications on Linux servers to suit the customer's needs. However, if I was to choose my favourite open source project, it would have to go to WordPress". It's not hard to agree with the reasons why. "It allows you to implement themes and modules that have been created by other developers and (kindly) shared with the community," says Craig. And the results are frequently extremely impressive.

Also consider:

DIASPORA www.joindiaspora.com
If you're looking to put together some kind of focused social networking service, then Diaspora is a personal web server that tackles that sort of job. The masterplan, and it's an ongoing goal, is to decentralise social networks, very much with an eye on better privacy. It's an intriguing project, currently in preview, but an ambitious one.

STATUSNET www.status.net
Designed with the idea of bringing businesses together via social software, StatusNet allows you to put together a specific social network system for an organisation. Not dissimilar to commercial products such as Yammer, it's a targeted communication project that claims to offer world-beating flexibility.

IDENTICA identi.ca
Identica, though, is more about microblogging (and it's been built using the aforementioned StatusNet). Or, in more digestable terms, it's a Twitter rival. It's not a massively popular Twitter rival right now, but open source has generally built from proverbial small acorns…

GWIBBER www.gwibber.com
Also for microblogging, why not give Gwibber a run? It supports lots of social networks and lets you amalgamate a stream from them. Aand, as you might expect, other features are on a take-it-or-leave-it basis.

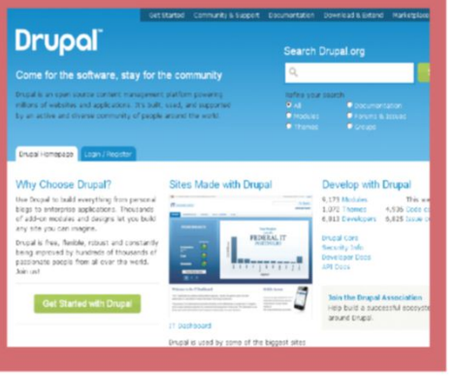
CONTENT MANAGEMENT SYSTEMS

The advent of the CMS has been a great leveller in web publishing, and open source has led the charge with a trio of exceptional tools.

DRUPAL www.drupal.com
Drupal is becoming increasingly adopted across the world, and it has no shortage of fans. Jon 'maddog' Hall is one, saying: "I think it is great when a project uses its own tools to sustain the project." It's used so far to build major websites, small blogs and pretty much everything in between, and estimates suggest that are many millions of sites and tools that have been built using the Drupal system. It's even used now to power The White House website.

JOOMLA www.joomla.org
Of equal profile is Joomla, which ticks many of the same boxes as Drupal. It, too, is an open source content management system, and it's claimed that it "powers 2.7% of the entire web". We've not counted, so we'll have to take their word on it. The core of Joomla has, of course, been heavily extended via plug-ins and retains a fundamental ease-of-use. Along with Drupal, it's showing how open source is providing the most flexible, interesting content management systems available.

APOSTROPHE trac.apostrophenow.org
If you're looking for an alternative, though, do consider giving Apostrophe a few minutes of your attention. Again, it's keeping the learning curve low that's the core of the project's aim, and while it lacks the sheer impact of Drupal or Joomla, Apostrophe is a strongly supported, able CMS.



ENTERTAINMENT

Contrary to popular belief, recreation and open source do go hand in hand, and there are a few projects that ably prove this.

FLIGHTGEAR www.flightgear.org
FlightGear is, bluntly, a quite brilliant flight simulation, working across many platforms and with a vibrant community behind it. It's ironic that Microsoft appears to have long since abandoned the flight simulation market, and that open source has proven – not for the first time – that there's enthusiasm, skill, quality and interest, in a manner that the bottom line of a spreadsheet will never demonstrate. It's ample evidence of knowing the price of everything and the value of nothing. Even if you're never tried a flight simulation before, it's worth preparing to be impressed by FlightGear.

VSCUMM www.scummvm.org
Emulators aren't in short supply in the open source world, although many of the popular ones (such as MAME, for instance) have restrictions that you might not think they have. ScummVM, though, is a smashing little tool, and available under the GPL. It allows you to play classic adventure games of old (which you do need to own), doing so by replacing the executable file and allowing them to run on platforms such as Linux. It's clever, broadening its game support and is a nostalgic's dream.

UAE www.amigaemulator.org
The ROM problem is also evident with the Universal Amiga Emulator, a sophisticated open source emulator. UAE, as it's known, isn't the friendliest of applications to use, but it can bring the highs and lows of the Amiga to your desktop.

CALIBRE calibre-ebook.com
If a good book is a better relaxant for you than a good game, then Calibre comes recommended. It's a useful utility for managing your library and syncing it to appropriate eBook readers. As eBooks continue to grow in popularity, so, we would suggest, will this.

"My favourite open source project? It would have to go to WordPress"
LUD reader Craig Jones

WEB BROWSING

MOZILLA FIREFOX

Website: www.mozilla.org
Get involved: www.mozilla.com/en-US/about/participate/

Whatever your personal feelings towards it, Mozilla's Firefox web browser was a massive mass-market breakthrough project for open source. More than any other application of its ilk to date, it's taken a significant chunk of market share off an established, previously dominant rival, and snared 31.24 per cent of the market (according to StatCounter figures). That's a staggering success, by any measure. Especially when you factor in the size of the market that we're talking about.

Firefox was very much the right project at the right time. Born out of the Mozilla project (which in turn was supported by Netscape), Firefox offered a slimline, clutter-free and effective browser that proved to be both fast and scalable. A series of fairly swift updates tuned it, and Microsoft's Internet Explorer consequently began a long decline that it's still struggling to arrest.

Firefox has reached such critical, cross-platform mass, arguably more than any other project we're chatting about here, that it has itself attracted some backlash. The light browser of old has clearly put on a little weight, while new browsers are suddenly looking a lot more innovative and modern than Firefox.

Yet Firefox has, in its comparably short life span, proven adaptable, popular and strong. And with Firefox 4 offering both a clearly Chrome-influenced design and a new, swifter update process, it'd be folly to write off one of the most important open source projects to date.

A shout, too, for Firefox Sync, incidentally, from Frederic Crozat at Novell: "Firefox Sync is probably the feature I like most from Firefox. It allows you to automatically synchronise your bookmarks, passwords and cookies across your computers, including your Android phone or tablet. Even better, only you are able to access your data stored on Mozilla servers."

Also consider:

CHROMIUM www.chromium.org
Google's Chromium project is having ramifications for web browsers, with the look and feel of Firefox 4, for instance, closely resembling Google's Chrome browser. Chromium is where Google pulls its code for Chrome, although it's a different entity. Chromium is also underpinning Google's plans for its Chrome OS operating system, too.

WEBKIT www.webkit.org
WebKit, too, as recommend by **Linux User & Developer** reader Mat Toor, is definitely worth checking out. Put simply, and in the words of Mat, "it's made mobile and Mac browsing amazing".

PRIVOXY www.privoxy.org
If you want a bit more privacy when browsing, Privoxy is a customisable way to remove adverts, overrule heavy caching and bypass censorship. As Frederic Crozat says, "Being able to remove all kind of advertisements, nasty pop-up and tracking cookies independently from your browser makes Privoxy a must."

KONQUEROR www.konqueror.org
It sells Konqueror short to list it solely under web browsers. However, many of us find Konqueror, which is designed as a file manager for the K desktop environment, a really solid way to browse the web. It's versatile and it's a welcome option.

SECURITY

CLAMAV & AVAST

Websites: www.clamav.net/lang/en/
www.avast.com/linux-home-edition

Anti-malware software isn't a massive concern on open source platforms, but the security software available for day-to-day Linux users is impressive. Two of the leaders are ClamAV (**www.clamav.net/lang/en/**) and Avast! (**www.avast.com/linux-home-edition**). Both will do battle against Trojans, malware, viruses and threats, but the latter is a lot more conditional than the former. ClamAV is fully open source, while Avast! Linux Home Edition is not available for any institutional use: it's strictly for home users only.

PROFESSIONAL TOOLS

One area where open source continues to excel is in developing projects that cater for very specific niches. What's most satisfying about this is that these are the areas where commercial entities have had carte blanche to charge what they like in the past, yet they're now being pushed by keen competition from the community.

ORANGEHRM www.orangehrm.com
OrangeHRM is a perfect example. This is an open source project that offers a full human resource management system. It's intricate, careful software that has to do a very tricky job. It allows organisations to chart absences, attendance, recruitment, benefits and suchlike, and over a million organisations worldwide are claimed to be using it. It is, clearly, a very niche piece of software. But for those who use it? It's saving them a lot of money, and doing a business-critical job– ironically, one that more and more people are less inclined to pass on to commercial software.

THE BIGGIES

ANDROID

Website: www.android.com
Get involved: developer.android.com

It's a name that came up time and time again during the compilation of this feature, with recommendations coming from writers, readers and prominent members of the open source community. With good reason, too. Udu Ogah, for one, puts it nicely. "It's brought premium mobile OS features to the masses. What was once reserved for the well-off is now accessible by millions thanks to it".

And that's the crux of just what Android has managed to achieve. In a market that was seemingly going to be dominated by closed-source practitioners such as RIM, Apple and Microsoft, the Google-backed Android has come from humble beginnings to become to the fastest-growing operating system on the planet. And as its popularity on mobile platforms grows, it asks some interesting questions as to the future of Microsoft and Apple's monopoly. For Android is now the most popular operating system on new smartphones in the US, and were Google to extend it to the desktop in some form (replacing its stalled Google Chrome OS project), then there's a real sporting chance that the old monopolies might be in serious trouble.

Whether that happens or not, Android is nonetheless a terrific, user-focused piece of software. Intuitive to us, and widely adopted, it may yet prove to be the most important and popular piece of mass-market open source work in the next decade. In the words of Jon 'maddog' Hall, it's "a great phone OS, and something to keep Jobs awake at night".

UTILITIES

CLONEZILLA

Website: clonezilla.org

Get involved: clonezilla.org/donations.php

Who needs Norton Ghost? While Symantec continues to pile on features in an effort to justify an annual purchase price for its software, Clonezilla has proven to be a tough, pretty much unbeatable adversary. It's one of the most mature, and strongest, open source utilities, and it's available in live and server guises.

It's job is an important one, and it makes little bones about getting on with it. For Clonezilla, as its title gives away, is a disk cloning application that, at its most extreme, can simultaneously clone over 40 machines, and it demonstrates real efficiency in the way it restores drives, too.

It's the speed of the program that's arguably its most impressive facet, and given how scalable it is (with a plethora of options and genuine cross-platform support), it puts the likes of Ghost to shame – and would do even if it had a price tag attached to it.

Also consider:

FILEZILLA filezilla-project.org
FTP is digested, streamlined and made breathtakingly easy with FileZilla. Tucking away no shortage of advanced features under its friendly exterior, it's a powerful and fuss-free project, and one of the most popular of this feature.

7-ZIP 7-zip.org
7-Zip makes a mockery of the continued attempts by the likes of WinZip to charge top dollar for file compression. It offers broad support, fast operation and works across many, many compression formats.

TRUECRYPT www.truecrypt.org
For quality, on-the-fly encryption, TrueCrypt comes recommended by many, ourselves included. It supports virtual disks, transparent encryption in real-time and also embraces the hardware opportunities offered by multi-core processors.

RECOVERY IS POSSIBLE www.tux.org/pub/people/kent-robotti/looplinux/rip/
We include Recovery Is Possible, meanwhile, as an example of how an entire strand of a distro can be targeted at a precise job. In this case, it's a very focused, suitably small distribution, one that might just prove invaluable in an emergency.

CONNECTBOT code.google.com/p/connectbot
Koen Vervloesem recommends ConnectBot as a useful little utility. "Thanks to this SSH client for Android," he tells us, "I can log into all my machines from everywhere with my phone. It also came in handy when I installed my headless NAS in the attic: no need to test the connection with a heavy laptop!"

PARTED MAGIC partedmagic.com
You get quite a collection of little tools with Parted Magic (including recovery tools), but its overriding purpose is to help you partition your drives. It does this by booting to a small, live distro, either via optical disc or flash drive. It's straightforward to use and fuss-free in its work.

WIPE wipe.sourceforge.net
If you're looking to erase data entirely from a hard drive, though, try something like Wipe. It's a secure file-wiping tool that uses multiple, intricate passes to ensure that once a hard disk has been erased, it's going to take something quite sensational to bring back any semblance of the data on it.

OPENBOX WINDOW MANAGER www.openbox.org
Openbox Window Manager is a neat tool. It's a window manager (but you probably guessed that) that's quick and speedy, and gives you plenty of control over how the windows on your desktop work. It's got lots of ideas and, while quite low-profile, is worth checking out.

FWBACKUPS www.diffingo.com/oss/fwbackups
Still, if all you're looking to do is take backups of your data, then open source offers a plethora of tools. Fwbackups is a very good one, ticking off all the key features you'd expect. Thus, scheduling, remote backups and ease-of-use are all high on the agenda.

MYSQL www.mysql.com
A substantive contribution to the software world, MySQL is the most popular open source database on Earth. As such, it's used by many very, very familiar names. Facebook, Google and Wikipedia are among the MySQL users in the world, and it also forms the backbone of many open source content management systems. There are paid editions available, but there's a GPL 2-covered release, too.

DEVELOPMENT

There are plenty of small tools that are of great benefit to developers, and we've had plenty of suggestions for those. Here's a selection of your recommendations...

ZSH www.zsh.org
"Since the day I discovered Zsh," says Koen Vervloesem, "I never looked back at Bash or any other shell. Zsh has a steep learning curve, but if you take the time to learn the basics, you'll become more productive in your shell sessions. Almost anything in Zsh can be tailored to your taste."

WIRESHARK www.wireshark.org
Koen is also one of many who has been keen to recommend Wireshark. "I use it to diagnose network problems daily. It's really indispensable for this purpose, as in many cases it allows you to see whether the problem lies at the client or the server side and what the underlying issue is. Wireshark has tons of features, like filtering, colourisation, graphs and so on."

VI www.vim.org
"Vi is an incredible tool," says Vincent Untz of OpenSUSE. "Its interface is completely obscure, with a horrible learning slope. And yet there is a point where it suddenly all makes sense and it feels completely intuitive, to a point where it makes you more productive than any other text editor. I'll say it publicly: I am a Vi addict, and I sometimes press Esc when I type text in other editors!"

BUGZILLA www.bugzilla.org
Bugzilla is potentially an invaluable project as well. It is server software that, as you might correctly conclude, lives and breathes to hunt down bugs. It does have broader capabilities than that, mind, and it's been deployed in both commercial and open source projects. For its job, it's hard to beat.

GCC gcc.gnu.org
GCC is vital to many developers, too. A popular GNU compiler collection, and one boasting front-ends for several languages, it benefits enormously from the speed and quality of its updates. And it's one of the finest open source compilers out there.

RUBY ON RAILS rubyonrails.org
Ruby on Rails is no slouch, either. And what's particularly appealing about it is that it manages to lower the often-considerable barriers to entry for programming. It speeds up development work for more experienced programmers, and offers a way in for those looking to get to grips with creating tools of their own.

GNU EMACS www.gnu.org/software/emacs
GNU Emacs is a mighty useful piece of software, because it lets programmers do lots of things from the one place, to the point where many who really take to GNU Emacs rarely leave it. So, you can plug in a mail and news reader, a project planner, a debugger and a whole lot more.

ECLIPSE eclipse.org
A development environment that works in a slightly different way to some, by utilising plug-ins on top of its runtime system for its main functionality, Eclipse as a consequence is very light at heart. What you need, you plug in – and what you don't, you don't. Simple.

SELENIUM seleniumhq.org
Finally here, Selenium is a suite of tools that's been specifically put together for testing web applications. Working across many browsers, operating systems, languages and frameworks, it boasts many individual projects as part of its comprehensive suite of tools.

WEB

APACHE

Web: httpd.apache.org/
Get involved: httpd.apache.org/dev/

"Without Apache," says Jon 'maddog' Hall, "very little could be achieved". That's no understatement, and he also adds that "Apache also was one of the first 'real' applications that made GNU/Linux into a world-class server machine, replacing a lot of SPARCs and Solaris systems."

At heart, Apache is web server software that played a pivotal role in the rise of the world wide web as we know it. It's the most popular HTTP server in use on the planet, and has been for 15 years and counting, serving over half of all websites.

Apache offers lots of things to lots of people. Many salute the fact that it's one of the fastest web servers in the world, while others appreciate its broad support. Immensely configurable, and the epitome of something that gets on with its job with little quarrel, Apache is vital software that just happens to be open source.

Also consider:

MYBB www.mybb.com
If you're looking to run bulletin boards, MyBB is what you need. **LUD** reader Connor Roberts recommends it for its "frequent updates and simply amazing community".

PRESTASHOP www.prestashop.com
On the eCommerce side, PrestaShop has built itself a strong and solid reputation. "It's clean and brilliant for developers," argues **LUD** reader Jason Holt.

RSSOWL www.rssowl.org
Few news readers offer the light footprint, speed and polish of RSSOwl. There's nothing in the commercial world to touch it.

QBITTORRENT qbittorrent.sourceforge.net
BitTorrent clients are a useful way to swiftly acquire the large files associated with Linux distros. Note that a lot of the big, free clients are closed source, but you're on safer ground with something like qBittorrent, which aims to be an open source alternative to uTorrent.

OPENSSH www.openssh.com
OpenSSH can be a vital security tool when online. It encrypts web traffic, including passwords, and also offers authentication and tunnelling capabilities.

DESKTOP ENVIRONMENTS

GNOME

Web: www.gnome.org
Get involved: www.gnome.org/community/

"One reason I love the GNOME project," says openSUSE's Vincent Untz, is because it innovated and pushed back limits so many times: the early focus on usability and on accessibility, the choice of the development model (the famous six months cycle that has been widely adopted by many other projects), the guaranteed stability of releases even during the development. In many areas, GNOME might not have been the first one to adopt a position, but it clearly made those philosophies much more popular in the free software world."

GNOME wasn't the forerunner in the desktop environment world, but since its inception it's certainly made a lot of the running. It's also emerged as the default environment in the majority of Linux distributions, including prominent distros such as Ubuntu, CentOS, Debian, Fedora and Red Hat.

It's not always had the easiest of times over the course of its development, but GNOME has emerged and thrived for two reasons. Firstly, the desktop environment itself is incredibly friendly and modern. And secondly, the development platform that it goes hand-in-hand with is exceptionally strong.

Also consider:

KDE PLASMA WORKSPACES plasma.kde.org
While there's more to open source desktop environments than GNOME and KDE Plasma Workspaces (to give the latter its proper title), the fact remains that most opt for one or the other. Plasma Workspaces is the umbrella term for KDE desktop environments these days, and you can find it installed by default on distros such as Gentoo, MEPIS and openSUSE. There are one or two political issues over time that have held some back from following KDE's path (indeed, these led to the creation of GNOME), yet there's little doubting the quality of what Plasma Workspaces offers. No wonder it's been a tough battle for desktop environment supremacy...

LXDE lxde.org
Speed is of the essence for LXDE, a lightweight and energy-saving desktop environment. It's conservative in its CPU and RAM use, and has cloud computing very much in mind. It works with many different flavours of Linux distro, too.

XFCE www.xfce.org
For an alternative lightweight desktop environment, you might also want to give Xfce a go. It tries to keep itself visually attractive while minimising system drain. And it does a fine job, too.

THE LINUX KERNEL

Website: www.kernel.org

Where do you start in trying to get across the significance of the Linux kernel itself? As Jon 'maddog' Hall says, "While not the first 'open' kernel, it helped launch the real blossoming of the FOSS vision. And of course, this was patterned after the UNIX OS and kernel created by Ken Thompson."

It was the Linux kernel, though, as conceived by Linus Torvalds at the start of the Nineties, that injected real impetus into the open source community, and was a major contributor to the movement. It embodies so much about what's great with regards to open source. The work of literally thousands upon thousands of contributors across the world, the Linux kernel is being developed on a daily basis, and itself stood on the shoulders of some terrific work done before its inception. But what it has subsequently laid a foundation for is for many more to stand on its shoulders, too, producing a variety of operating systems and software, all based around the same core. The kernel ensures that Linux, in its many guises, is the most diverse, pragmatic and evolving operating system on the planet.

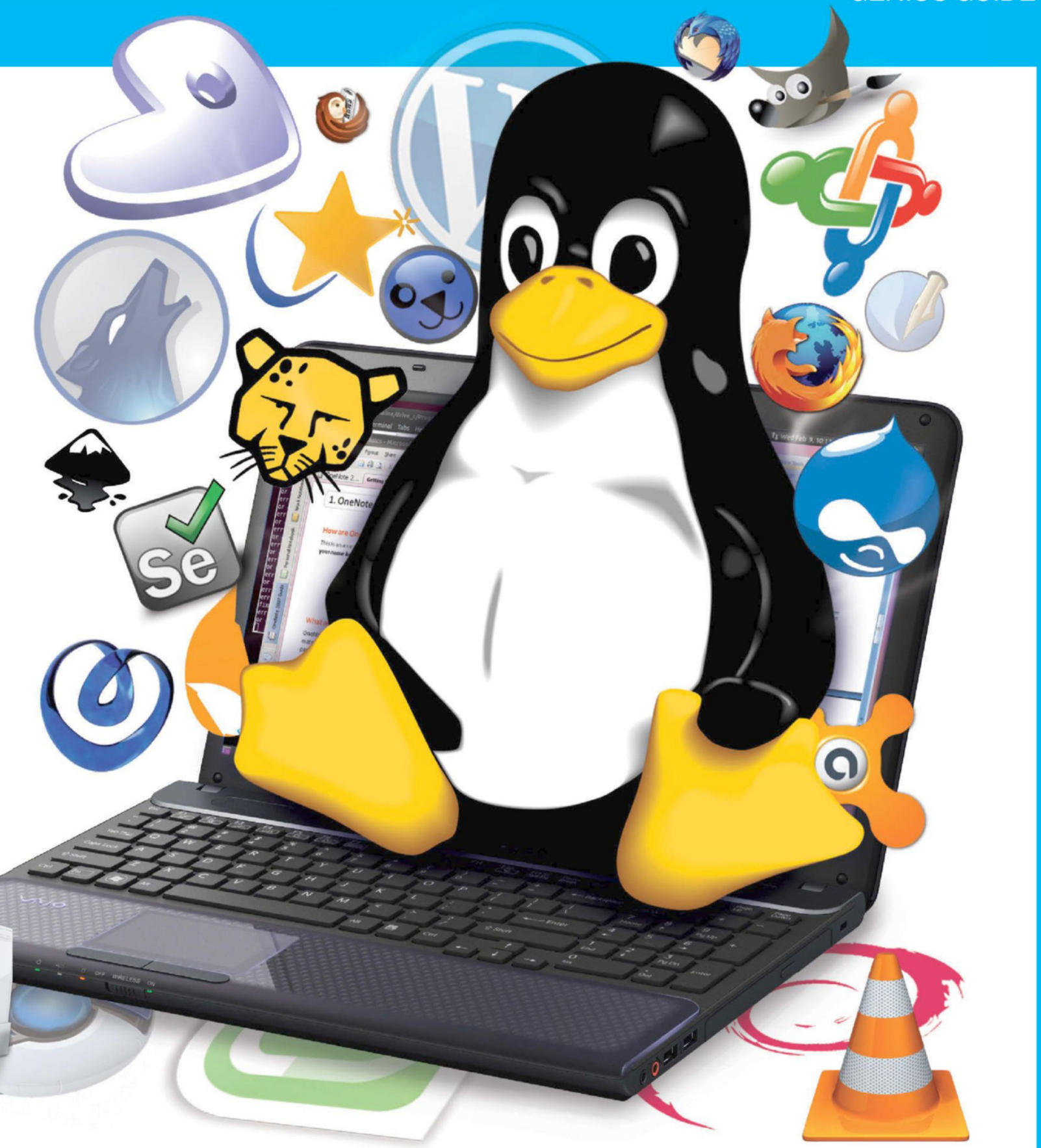
It's also a project that could only have happened with the goodwill and efforts of thousands of devoted volunteers. Various studies over the years have tried to put a price tag on what it would have cost a commercial developer to develop Linux to the point we find it at now, and the biggest estimated bill has been $3bn.

Yet the contributors to the project largely haven't been in it for the money. Instead, the kernel embodies a commitment from many towards genuinely free software. And with computing and technology adopted to different uses, form factors and degrees of portability, it's that ethos that's led to open source development, and the Linux kernel, being fastest to adapt to the big changes taking place in software.

Have you spotted the deliberate mistake? There are indeed only 99 projects listed. The last spot is reserved for one of the thousands of projects we simply didn't have the space to include. So why not let us know why your favourite deserves a place on our next list. You can reach the Linux User team via the website www.linuxuser.co.uk and email - linuxuser@imagine-publishing.co.uk.

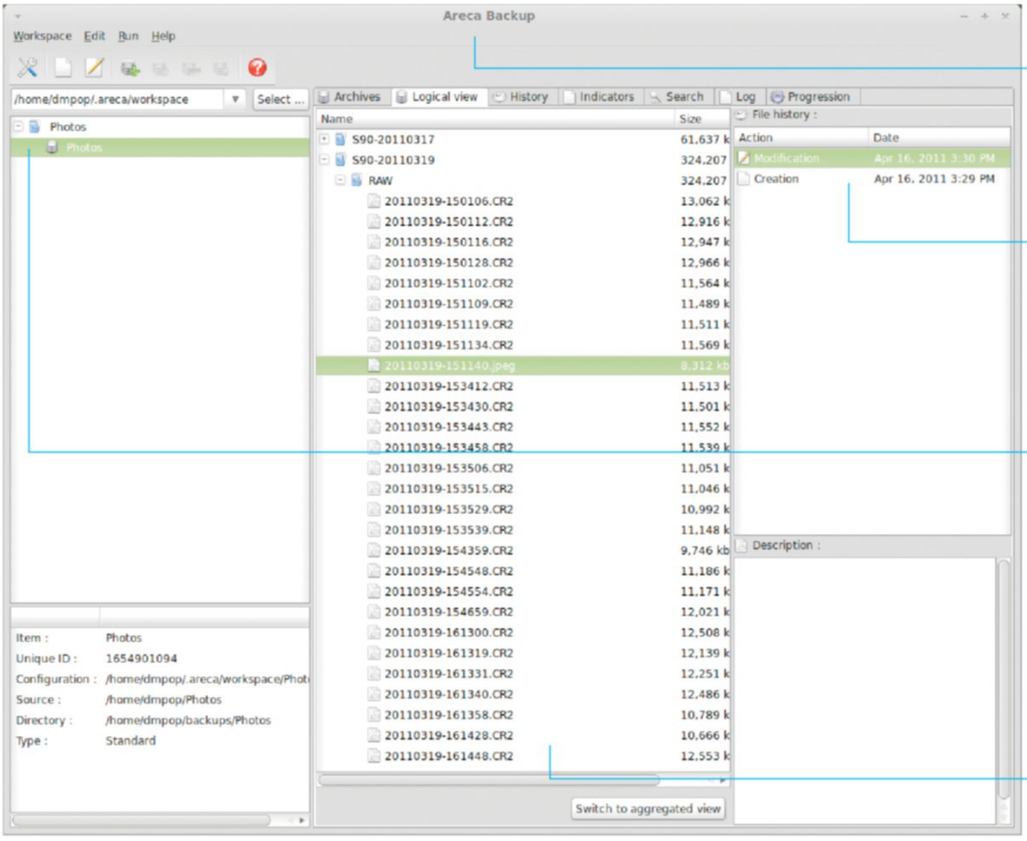
Tips & Tricks

Master essential Linux and Open Source skills


Se
1. OneNote

Never lose another file with Areca Backup

Keep your data safe the open source way. Set up an efficient on-site and off-site backup solution using the Areca Backup tool



Areca Backup sports an intuitive interface that puts all essential features at your fingertips

Areca Backup maintains a file history, so you can easily recover a specific version of a file

The application allows you to set up multiple backup profiles, so you can, for example, create separate backup jobs for different types of data

Using the Logical view, you can browse the backup archives and preview files

Advisor

Dmitri Popov has been writing about Linux and open source software for almost a decade, with a focus on productivity and collaboration software

Keeping your data safe can be a chore, unless you have a decent backup tool. While many Linux users rely on rsync or duplicity as their backup tools of choice, these utilities are not the only game in town. And if you are looking for a user-friendly and powerful backup application that sports both graphical and command-line interfaces, you might want to take a closer look at Areca Backup. This tool packs a lot of nifty features which can come in handy when setting up a backup solution. This includes support for data compression and encryption using the AES 128 or AES 256 algorithms, powerful filtering capabilities, as well as support for incremental, differential and full backup. Areca Backup can back up data to external storage devices, FTP and SFTP servers, and Samba shares. Better still, Areca Backup's functionality is wrapped in a user-friendly graphical interface, making it supremely easy to set up, perform and monitor backup operations. Here's how to get started...

Resources

Areca Backup www.areca-backup.org

An external storage device for on-site backups

An FTP server for off-site backups

1. Install Areca Backup

First off, make sure that the Java Runtime Environment (JRE) is installed on your machine. Then grab the latest version of the tool from the project's website (**Fig 1)** and unpack the downloaded archive. In the Terminal, switch to its directory and run the areca.sh script.

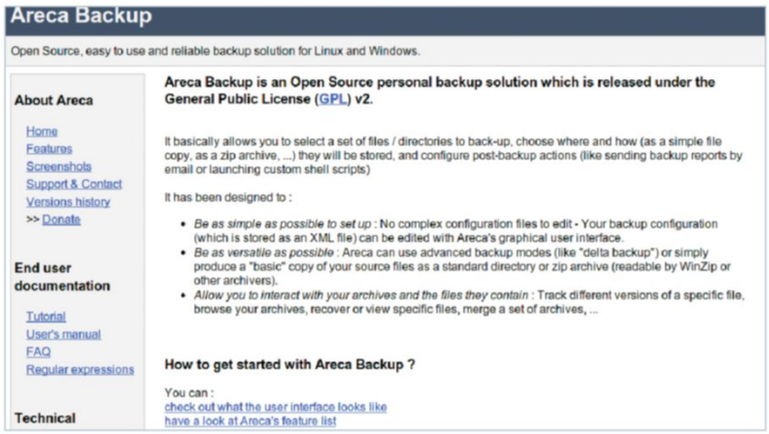
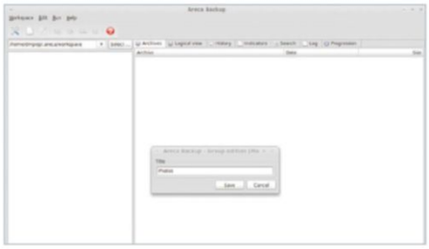


Fig 1 Install Areca Backup

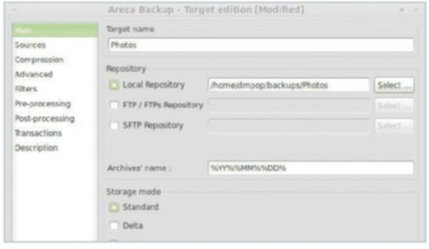
2. Create new group

Areca Backup allows you to create multiple backup jobs called groups. For example, you can set up different groups for your documents and photos. To create a new group, choose Edit> New Group, or press Ctrl+G. Give the new group a descriptive name and hit OK.



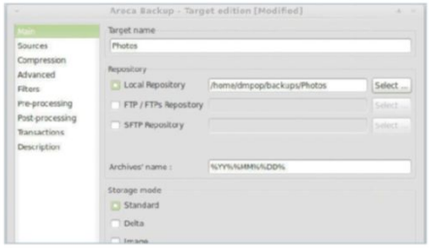
3. Set up new target: Main

Next, you need to create a backup configuration, or 'target' in Areca Backup's parlance. Choose Edit>New Target, or press Ctrl+T. In the Main section, give the target a name and specify a local or remote target directory.



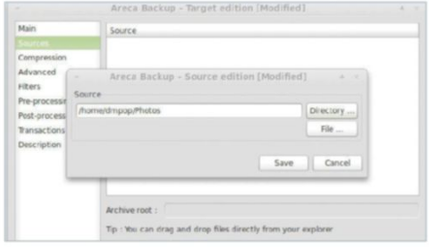
4. Set up new target: Storage mode

In the Storage mode subsection, select the desired backup type. You can choose between the Standard (a new archive is created for each backup), Delta (creates a separate backup archive containing modifications made since the last backup) and Image (a separate archive is created and updated at each backup).



5. Set up new target: Sources

To add the files and directories you want to back up, switch to the Sources section, and press the Add button. Select the desired directory or file and press OK. This way, you can add as many directories and files as you need.

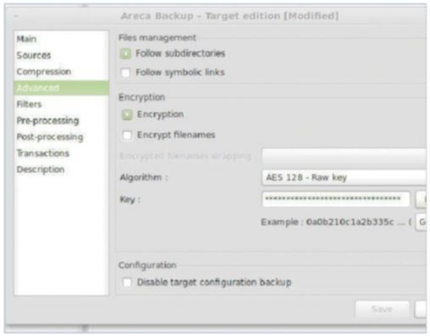


6. Set up new target: Compression

In the Compression section, you can enable archive compression and specify compression options. You can choose between different compression types and storage options. If you want to split backup archives into chunks of predefined sizes, you should enable the Splitting option.

7. Set up new target: Advanced

To enable encryption, switch to the Advanced section and tick the Encryption checkbox. You can then select the algorithm to use. In the File Management subsection, specify how the backup should treat symbolic links and subdirectories.



8. Set up new target: Filters 1

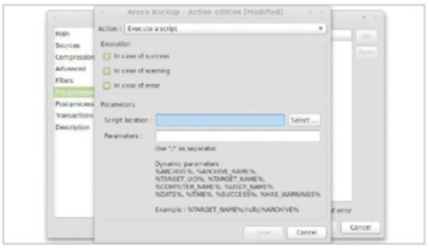
In the Filters section, you can specify rules that include or exclude files matching certain criteria from the backup. To create a new rule, press the Add button and select the desired filter type from the Filter type drop-down list. For example, if you want to exclude files with certain file extensions, select the File extension filter item and tick the Exclusion filter checkbox. Then enter the file extension you want in the Parameters field (eg .TIFF).

9. Set up new target: Filters 2

Areca Backup supports a wide range of filter types. Using the Regex filter, for example, you can create advanced exclusion rules, while the File date filter lets you exclude files older than a specified date. You can also exclude locked files and files that are owned by a specific user.

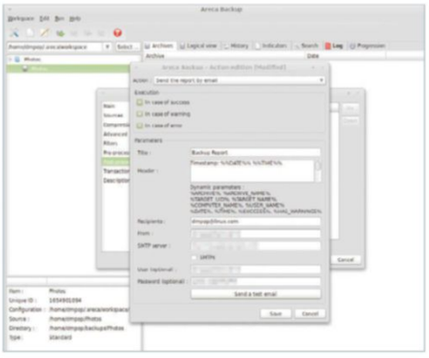
10. Set up new target: Pre-processing

Using options available in the Pre-processing section, you can specify actions that run before the backup operation. You can configure the pre-processing options to run a script or to delete older backup archives.



11. Set up new target: Post-processing

In the Post-processing section, you can specify actions to run after the backup operation is completed. In addition to scripts, Areca Backup lets you save customised reports on the disk or send them by email. The latter option can come in handy to keep tabs on backups when you are out and about.



> **Keeping your data safe can be a chore, unless you have a decent tool like Areca Backup**

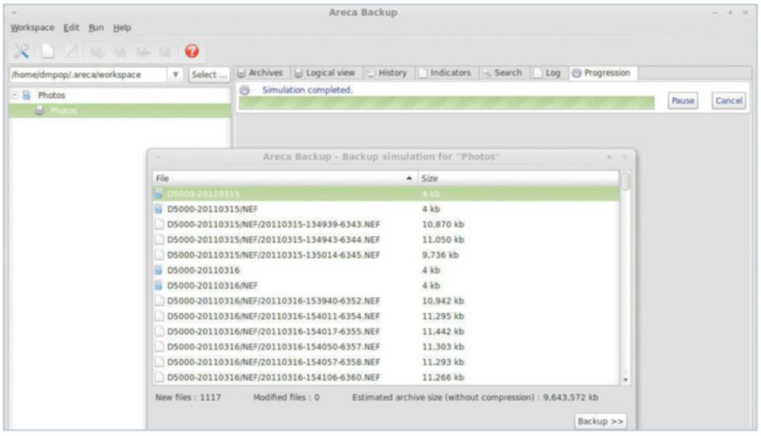
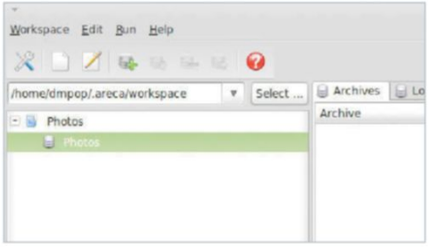


Fig 2 Backup simulation

12. Save new target

When you're done configuring the options, press the Save button to save the changes and close the dialog window. You should see the new target in the group you created earlier. To edit the target later, either double-click on it or use the Ctrl+E keyboard shortcut.

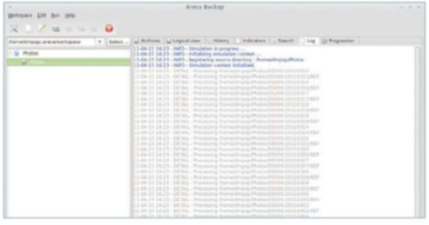


13. Perform a dry run

To make sure that the created target works properly, it's a good idea to do a dry run. Right-click on the target and choose the 'Simulate backup' command (or use the Ctrl+S shortcut). Wait till the operation is completed and then close the Backup simulation window (**Fig 2**).

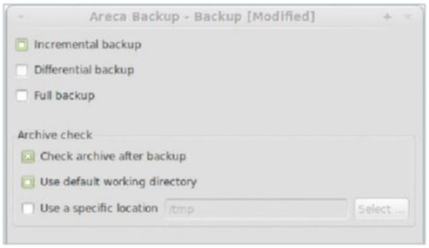
14. Check log for errors

Click on the Log tab and check the log for error messages. If everything went smoothly and there are no errors in the log file, you are ready to perform a backup.



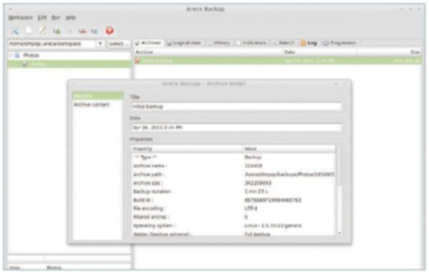
15. Run backup

To run the backup, right-click on the target and choose the Backup command (or press Ctrl+Enter). In the dialog window that opens, select the desired backup type. While you can leave the rest of the options at their default values, you might want to provide a descriptive name and a short description of the current backup, so that you can easily identify it later. Press the 'Start backup' button to run the backup operation. You can monitor the progress in the Progression section.



16. Examine backup archive

Once the backup operation is completed, you can see the newly added backup archive under the Archives tab. Double-click on the backup archive and you can then view detailed info in the Manifest section.

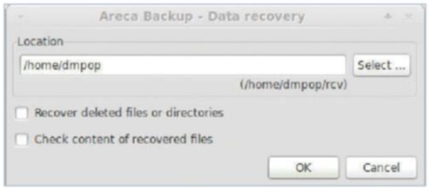


17. Quickly restore individual files

You can use the Archive content section to browse the contents of the backup archive and quickly restore individual files. To do the latter, locate the desired file, right-click on it and choose the Recover command.

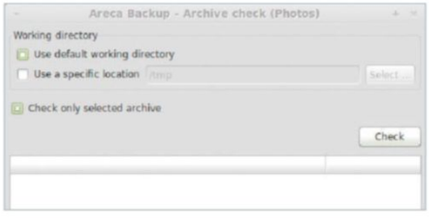
18. Restore entire archive

To restore the entire backup archive in one fell swoop, right-click in it in the Archives section and choose the Recover command. Specify the destination directory for the recovered files and choose whether you want to recover deleted files or check the recovered files. Hit OK to perform the restore operation.



19. Check backup archive

To make sure that the backup archive is in good condition, you should check its integrity. To do this, right-click on the archive, choose the Check command and press the Check button. Once the operation is completed, check the log for errors.



20. Logical view

You can use the Logical view section to explore the contents of the backup archive. When you select an individual file, you can see its versions in the File History pane (**Fig 3**). You can use the View and Recover commands to view and restore a specific version of the currently selected file.

21. Merge archive

To reduce the number of backup archives, you can merge them. To do this, select the archives you want to merge, right-click on the selection and choose the Merge command. Press the 'Start merge' button to merge the selected archives. Note that merging archives deletes all previous versions of files.

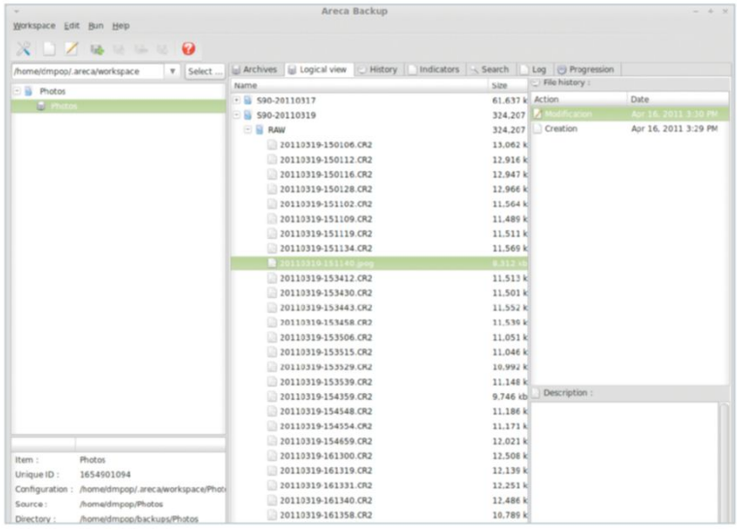
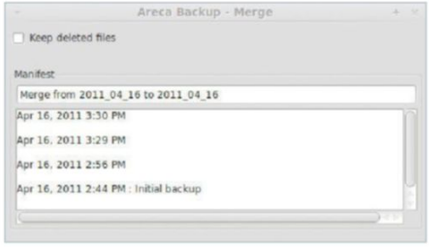
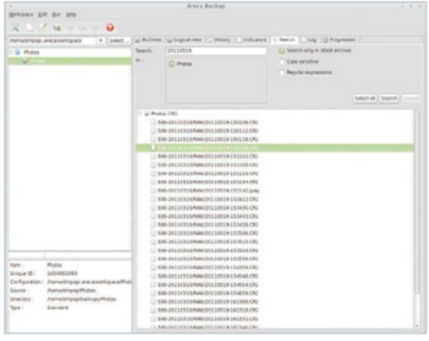


Fig 3 Areca Backup's Logical view



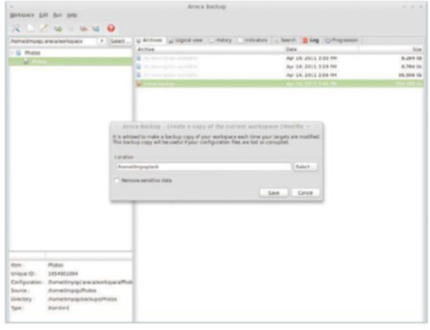
22. Perform searches

The Search section lets you locate specific files in backup archives. You can limit searches only to specific targets and the latest archive, as well as use regular expressions to construct advanced search queries.



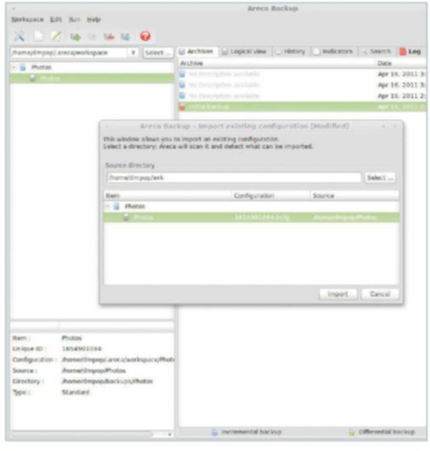
23. Backup workspace

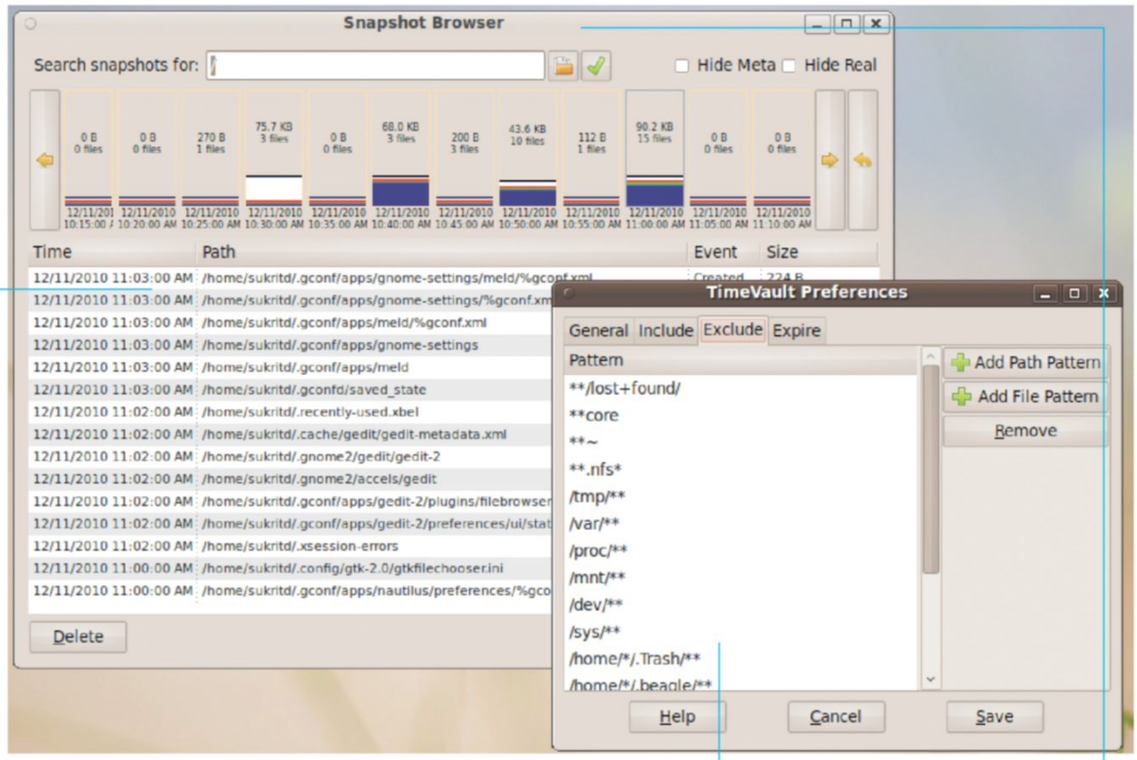
To avoid losing your backup profiles, it's a good idea to keep a backup copy of the entire workspace. To do this, choose the 'Save copy as' command from the Workspace menu and specify a location for the configuration file.



24. Import configuration file

Whenever you need to restore the workspace, choose Workspace>Import and select the directory containing the .bcfg configuration file. Select the file and then hit the Import button to restore the workspace.





The Snapshot Browser displaying all the backups in chronological order

The exclusion of unnecessary files makes your backups leaner

You can directly restore or open previously backed up files from the Snapshot Browser

Back up like a pro with TimeVault

TimeVault is a feature-packed backup and restoration system which will make your life a lot of easier and help you rest easy in the knowledge your data is secure

Resources

TimeVault http://launchpad.net/timevault

Advisor

Sukrit Dhandhania has spent several years working professionally, implementing several open source tools for companies. During this time he has evaluated, set up and maintained various open source tools for these firms

If you have used a recent Macintosh computer you have probably heard of Time Machine, the wonderful backup system bundled with Mac OS X Leopard onwards. TimeVault is a Linux front-end application for taking snapshots (ie copies, protected from accidental deletion) of your entire system or of sets of directories at a certain point in time. It also has a restore functionality which integrates quite nicely with the Nautilus file manager. Let's look at how to install and set up TimeVault on your computer.

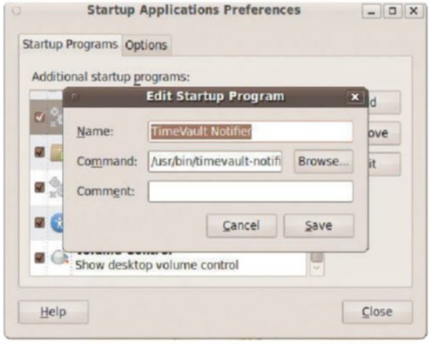
01 Install TimeVault

To install TimeVault on your computer, you will first need to visit the project's website and download the installer package (**Fig 1**). As of now, binaries are only available for Debian-based systems. Users of other distributions of Linux can use the source release. On Ubuntu, download the latest binary release from the project's website and open it with the Ubuntu Software Center. Install any dependencies that show up in the Software Center, and then install

TimeVault. At the time of writing, TimeVault did not support Ubuntu 10.10. It worked fine on Ubuntu 9.10, though.

02 Set up Notifier

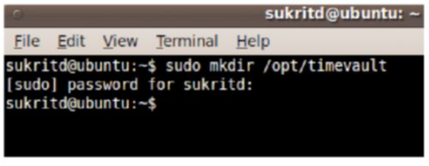
Once you have completed the installation of TimeVault, you should log out and back in. You will notice the TimeVault Notifier icon in your system tray. If it's not there you will need to set it up manually. In the GNOME system menu go to System>Preferences>Startup Applications. Under the Startup Programs tab of the window that opens, click on the Add button. Give the new startup item an appropriate name, such as 'TimeVault Notifier'. Under the command section enter the absolute path of the notifier – '/usr/bin/timevault-notifier'. Log out and back in for this setting to kick in.



Set up the Notifier for easy access to TimeVault's options

03 Create snapshot directory

Before you begin using TimeVault you will need to look on your hard drive and find a partition with enough space to be able to hold your backups. In this tutorial we're going to use the folder /opt for this purpose. Create a new directory under /opt called 'timevault' or 'backups'. Use the following command: '# sudo mkdir /opt/timevault'.



Create a directory in a partition that has a reasonable amount of space so you can have many backups

04 Configure TimeVault

Now you need to configure TimeVault and set up precisely what you want backed up. Launch the Preferences pane from the

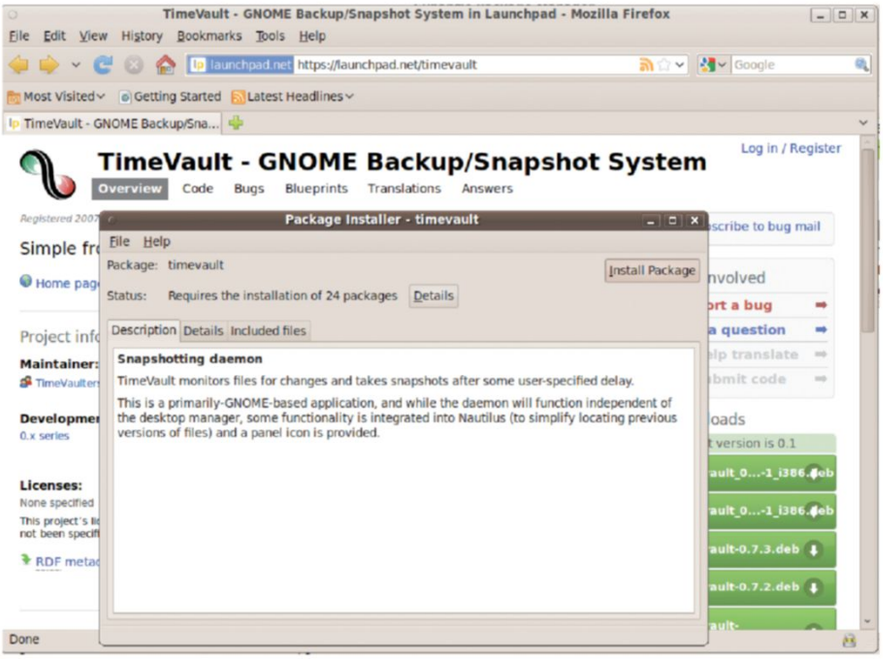
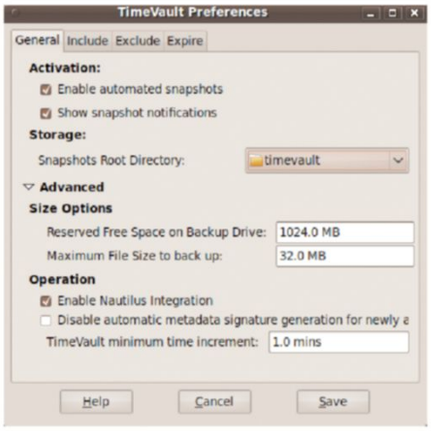


Fig 1 Install TimeVault The installation process for TimeVault

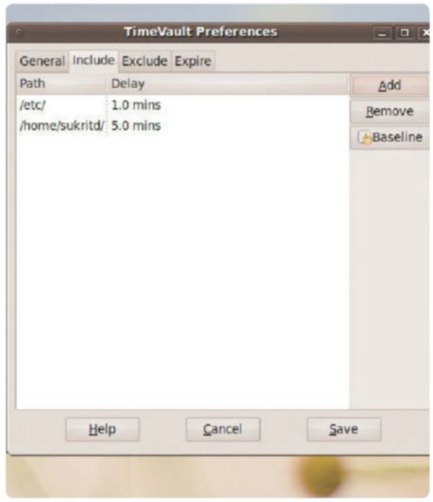
Notifier. Under the General tab, check the 'Enable automatic snapshots' and the 'Show snapshot notifications' options so that the backups are taken without your initiating them and you are kept informed when they are done. You should also set the 'Snapshots root directory' property.



Add the location of the backup directory to the TimeVault configuration pane

05 What to back up

Under the Include tab of the Preferences pane, you need to configure what directories you want backed up. You can add the important directories such as /etc and /home. You can customise this setting in the next step.

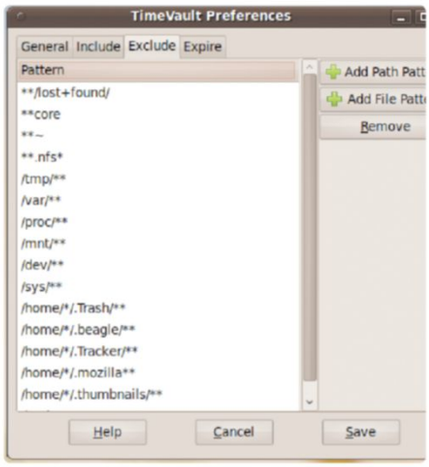


This step is very important because here you decide what you want backed up

06 Exclude subdirectories

In the last step, you set up some directories whose snapshot TimeVault will take. If you click on the Exclude tab, you can add files, directories and patterns that should be excluded from the backup. So if you want to exclude all

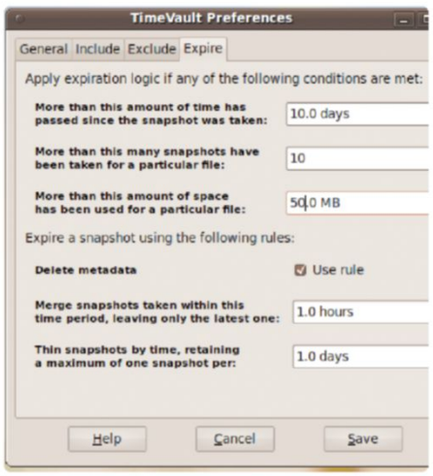
your NFS files, you need to make an entry for something like '**.nfs*'. You would also want to exclude directories such as '/mnt/**' and '/dev/**'.



It is important to tell TimeVault to not back up certain files or folders

07 Expire your snapshots

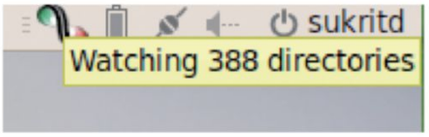
Under the Expire tab you can set the amount of time TimeVault to hang on your snapshots. This setting has three factors – amount of time, number of snapshots and the amount of space being occupied. This group of settings is quite useful as you don't want the same file backed up an unlimited number of times. A safe configuration is to set up the backup for '10.0 days', 10 copies and set a 50MB space limit. Make sure to hit Save when done.



Configure how long you want TimeVault to retain your backups

08 Backup status

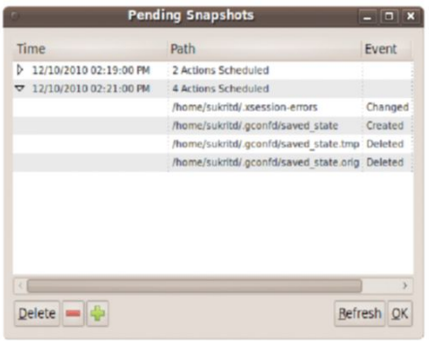
When you hit the Save button, the Preferences pane will close. Hover over the TimeVault icon in your taskbar. You should now see the progress that your first backup is making. The icon will let you view how many files are scheduled for backup and how many directories are being watched by TimeVault.



TimeVault displays how many files are pending backup

09 Detailed status

Left-click on the TimeVault icon to see a more detailed report of the update status. You will see a window displaying which files and directories are pending in the backup process. You can make a few small changes at this time, such as deleting a particular file from backup, but we would advise you against tinkering with an ongoing backup.



You can get a more detailed status of your backups from the Pending Snapshots window pane

10 TimeVault updates

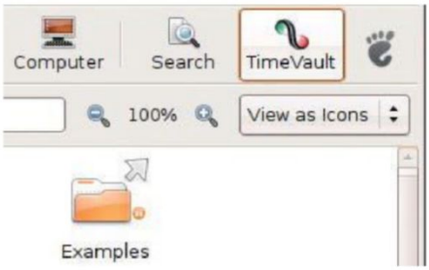
Whenever a file that TimeVault is watching has changed, the backup tool smartly detects it, notifies you from the icon in the taskbar and adds it to its list of files to be backed up at the next scheduled backup process.

11 The Snapshot Browser

Once TimeVault has taken a few backups, you will be able to see and access versions of the snapshots from the Snapshot Browser (**Fig 2**). The Snapshot Browser is a window that very handily displays the history of your snapshots and allows you to revert individual files as and when you want. This is where TimeVault gives you an added edge over many other simple backup solutions available. Right-click on the TimeVault icon in the tray and click on the Snapshot Browser option to access it. You can also use the 'Snapshot Browser as Root' option to restore files to which you might not have permission otherwise.

12 Snapshots in Nautilus

You can also access your TimeVault snapshots using the Nautilus file browser. When you launch Nautilus, you will find a TimeVault icon at the top of the window. Click on it to access your snapshots. We found that as of now, the integration with Nautilus has more features than the TimeVault Snapshot Browser.



The TimeVault icon in Nautilus

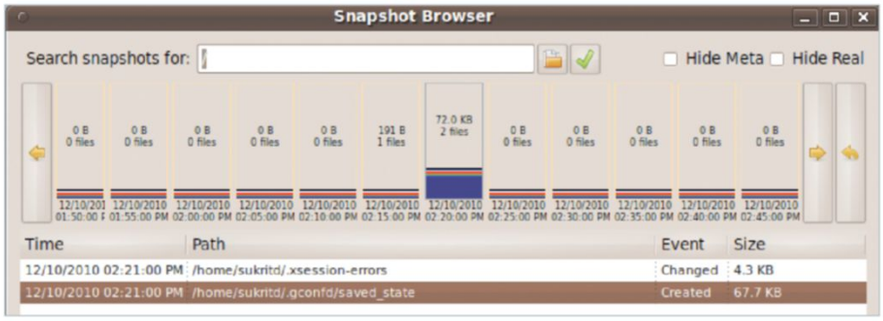


Fig 2 The Snapshot Browser TimeVault's handy Snapshot Browser gives you a visual representation of your backups

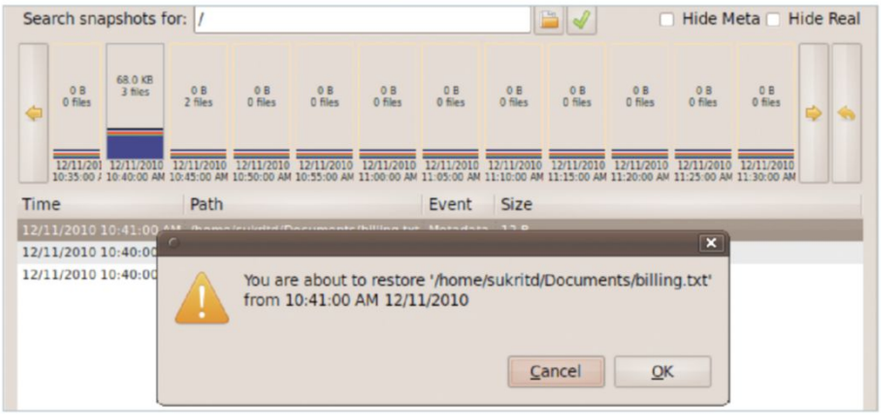
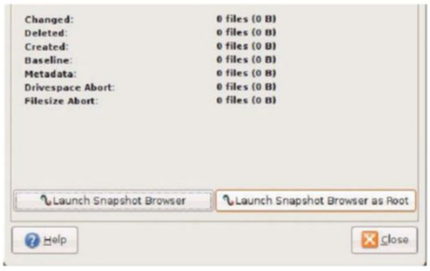


Fig 3 Restore Process Be careful during the restore process, otherwise you might restore the wrong version

13 Properties windows

There is a third way to access your file snapshots. Launch Nautilus and right-click on the file you want to restore. Click on the Properties option in the menu and select the tab with the TimeVault icon. From here you can view the history of the file's backups, and directly access it in the Snapshot Browser.



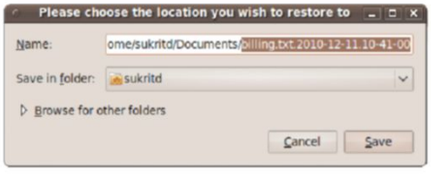
The Properties dialog box for the files backed up with TimeVault

14 Restore process

Easy restoration of your backups is where a program such as TimeVault can prove really useful. Launch the Snapshots Browser and locate the file or folder you want to restore, and find its right version. Select the file you want to restore. Now hit the Revert button. You will see a pop-up message asking you if you are ready to go ahead with the restore (**Fig 3**).

15 Name and destination

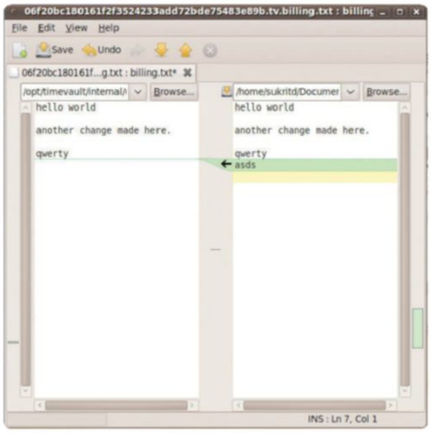
Now you'll be asked where you want the restored file to be placed, whether a folder or somewhere else. You will also have to decide whether you want the restored version of this file to replace the original file or be named something else.



You can place the restored file in a different location if you like

16 Viewing and Comparing Versions

When you select a file in the Snapshot Browser, you are offered some options at the bottom right of the window pane. You can choose to revert a file, as we learned in the previous step. You can also open the file to view it, or perform a 'diff' to compare the changes in versions.



A diff between two backups of the same file

17 Using an external hard disk

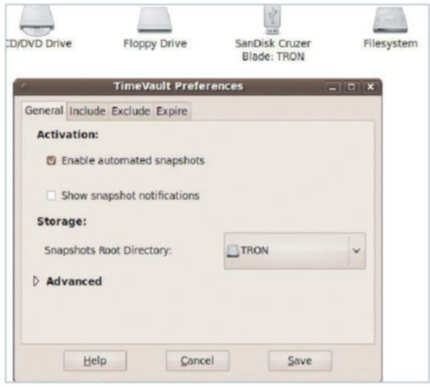
Ideally you do not want to use your laptop or desktop's hard drive to store your backups, which would defeat the purpose of making

"Store backups on an external hard drive"

them. You should store backups on an external hard drive or in a remote location. We set up TimeVault to store our system's snapshots to an external USB hard drive mounted under /media. All was well until the point where the hard drive wasn't plugged in any more. TimeVault failed to recognise this and backed up to that location anyway. This resulted in it starting backups from scratch and filling up our root partition.

18 External drive workaround

So far there's no simple way to work around the issue of using an external hard drive mentioned in step 17. We hope the TimeVault team releases an update with a fix soon. What you can do, though, is to stop the TimeVault daemon while your external drive is unplugged, and start it when it is plugged back in.

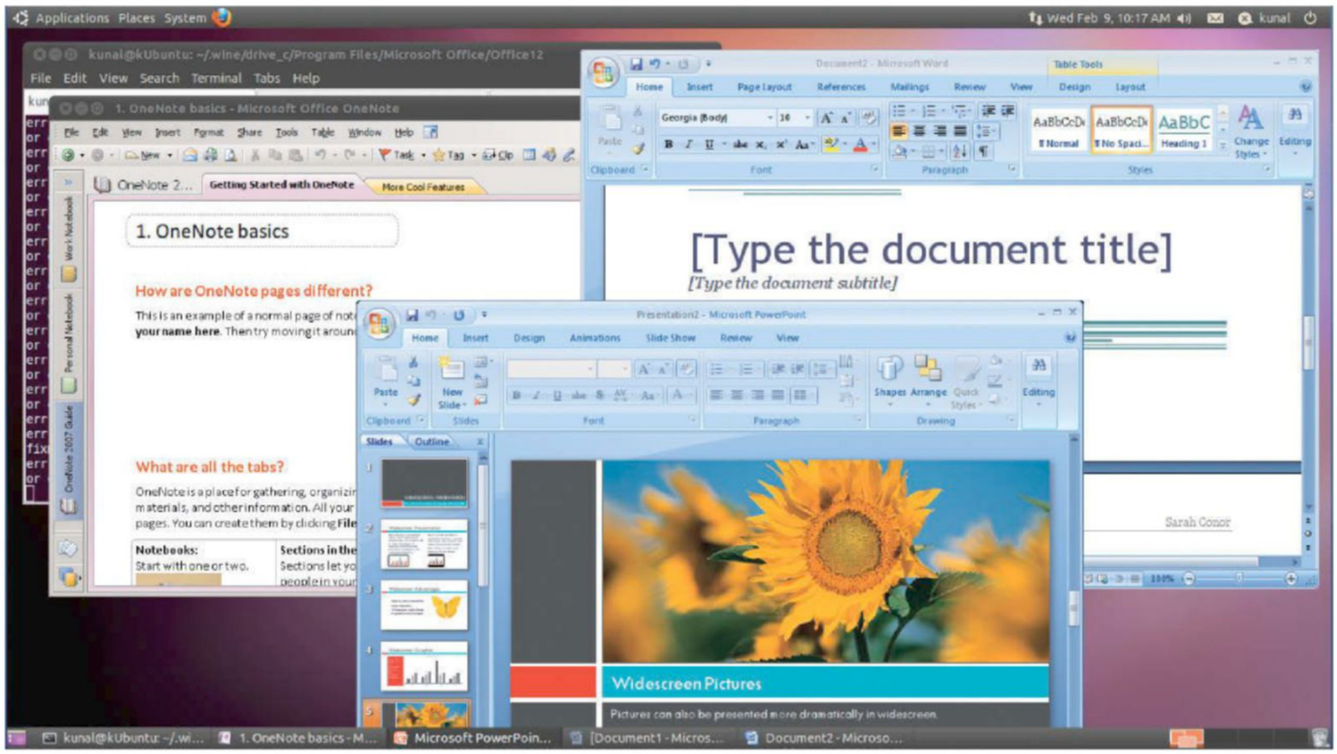


Backing up on an external drive is tricky

Overall, TimeVault is a great backup option for both advanced and relatively new users. It may be particularly good for users who have migrated from an Apple Mac computer. The software works great for most part, but there are certain features that still need to be included and others that need to be fixed. There are a few versions of TimeVault floating around that have these features and that could be included into the main releases. Still, TimeVault is shaping up quite well and like most open source programs, as the number of users grow, so will the development effort of the project.

Seamlessly run Windows software on your Linux system

Sometimes using Windows applications is unavoidable, so in this tutorial we will make use of Wine to show you how to run Windows applications seamlessly within Linux



Microsoft Office 2007 Applications running on Ubuntu

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Even though the Linux software catalogue is huge, there are times when you may feel that you are missing an application that is otherwise available on Windows. There could be many reasons behind why you want that exact application. Probably the most common is that support for a particular file format or an application used at work is simply not available for Linux. Thankfully, there are many ways by which you can use a Windows app on your Linux system. **Here's a quick breakdown of some of the most popular methods...**

Dual booting: In this method you will need to create a separate partition and install Windows on it. Then whenever you need to run a Windows application you will need to restart your system, then boot into Windows and use the app. This method is only recommended for an intermediate audience who are comfortable with partitioning their hard drives. It also requires that you have a large amount of disk space at your disposal.

Virtual machine: In this scenario you will need to install a virtualisation program such as VMware Workstation or VirtualBox on your Linux system, then install Windows as a virtual machine. You

> **Wine is a translation layer capable of running Windows applications on Linux and similar other POSIX-compatible operating systems**

can the use this virtual machine to run Windows applications without rebooting your system. However, using virtual machines requires a considerable amount of system resources in terms of RAM, CPU and disk space as the virtual machine needs to run a fully fledged operating system within another operating system simultaneously.

Using Wine: Wine lets you run Windows applications without rebooting or virtualisation. In this tutorial we will be using Wine to run Windows applications on a Linux system.

Introducing the world of Wine

Wine (a recursive acronym for Wine Is Not an Emulator) is a translation layer (or a program loader) capable of running Windows applications on Linux and similar other POSIX-compatible operating systems. Wine does not emulate Windows applications on Linux – instead it provides alternative implementations of DLLs that a typical Windows application calls and a process substitute for the Windows NT kernel. Wine is made of 100 per cent Microsoft-free code.

Wine supports a large number of applications, but not all are supported equally. You can visit the Wine Application Database (AppDB, **http://appdb.winehq.org**) to see how well your favourite Windows application works with Wine. AppDB is maintained by the community and you can also add your own discoveries. **AppDB defines the following type of ratings…**

Platinum: An application can be rated as Platinum if it installs and runs flawlessly 'out of the box'. No changes are required in Wine configuration files.

Gold: Application works flawlessly with some DLL overrides, other settings or third-party software.

Silver: Application works excellently for 'normal' use. For example, a game works fine in single-player but not in multiplayer; Windows Media Player works fine as a plug-in and standalone player, but cannot handle DRM etc.

Bronze: The application works, but it has some issues, even for normal use. For example, a game may not redraw properly or display fonts in wrong colours, be much slower than it should etc.

Garbage: An application gets this rating if it cannot be used for the purpose it was designed for. If so, there should be at least one bug report in Bugzilla. The application cannot be installed, does not start, or starts but has so many errors that it is nearly impossible to use it.

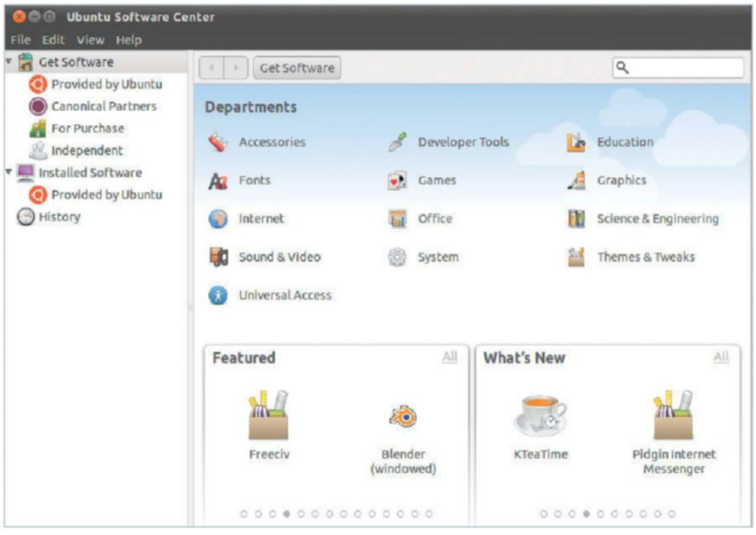


Fig 1 Installing Wine Ubuntu Software Center

01 Installing Wine

Wine is available for all popular UNIXes, including Ubuntu, Debian, Red Hat, SUSE, Mandriva, FreeBSD, Solaris and Mac OS X.

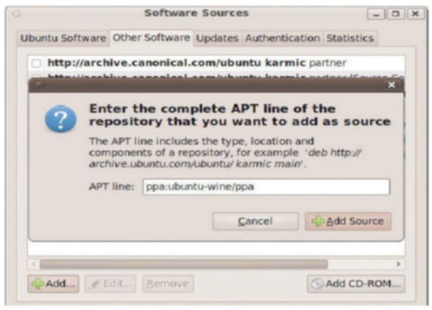
The following explains how to install Wine…

On Ubuntu:

1. Open Ubuntu Software Center by opening Applications>Ubuntu Software Center (**Fig 1**).

2. Then click on Edit>Software Sources. Select the Other Software tab and click Add. Then enter the following apt line:

```
ppa:ubuntu-wine/ppa
```



Managing software sources

Dual Booting vs Virtualisation vs Wine

Feature	Dual Booting	Virtualisation	Wine
System Restart Required	Yes	No	No
Operating System Licence Required	Yes	Yes	No
System Resource Required	Fair	More	Less
Application Compatibility	Excellent	Good (3D games and other demanding applications may not run properly)	Fair
Application Startup Time	Excellent	Fair	Good

3. Select 'PPA for Ubuntu Wine Team' from the Get Software Section on the left, then select Microsoft Windows Compatibility Layer (Development files) – wine1.3-dev. And click Install. This will install the latest Wine packages, including development libraries.

Tip: Uninstall the existing Wine packages for maximum compatibility. Also, for a clean install you can remove the Wine user directory by using the command 'rm -r -v .wine' from your home directory.

02 Installing Wine support packages

Now we need to install non open source (but free) support packages. These include packages such as Microsoft Core Fonts, Visual C++ runtime etc.

So let's get on with it...

Install Cabextract :

This package is required to extract Microsoft .cab files.

```
$ sudo apt-get install cabextract
```

Download winetricks script:

Now we need to download a nifty script called winetricks, which automatically downloads and installs the necessary support packages.

```
$ wget http://winezeug.googlecode.com/svn/trunk/winetricks
```

Now enter the following command to install the necessary support packages:

```
$ sh winetricks corefonts tahoma
```

Tip: You can also run winetricks without any command-line parameter so that you can select the packages you need to install from a nice GUI interface.

03 Installing Microsoft Office

Our system is now ready to accept Microsoft Office 2007. Insert the Microsoft Office 2007 disc and right-click on setup.exe and select 'Open With Windows Program Loader'. This should launch the Microsoft Office Installer. Install it in the usual way, as you would do on any Windows system.

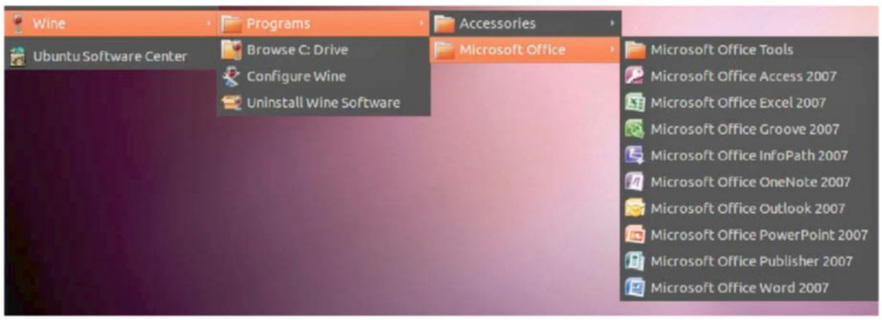
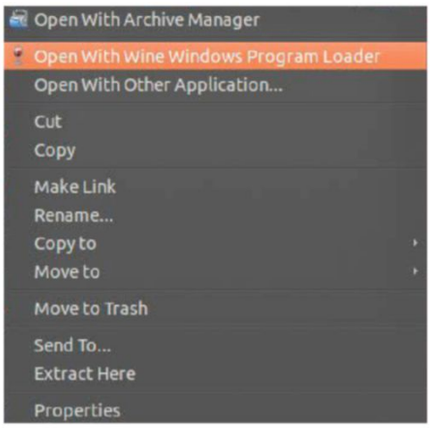
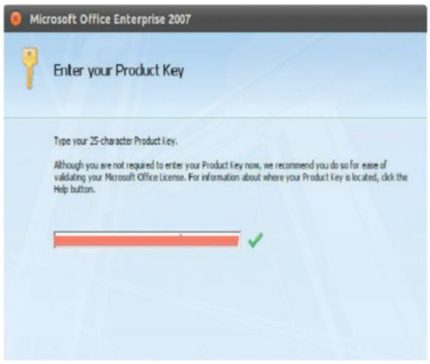


Fig 2 Running Microsoft Office Wine application menu



Wine program loader

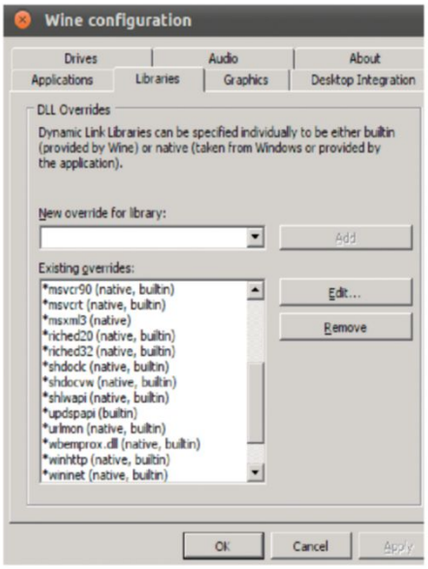


Office setup wizard

04 Setting up libraries

Now that we have installed Microsoft Office 2007, we need to make sure that Wine is configured to use it properly. You see, Wine comes with its own implementation of various Windows libraries, but we can use the native Windows libraries directly to make sure that applications have maximum compatibility. Wine comes with its own configuration tool called winecfg. Among other things, it also allows us to switch between the built-in and the native version of the libraries.

Open the Wine configuration tool by entering the command 'winecfg' . Go to the Libraries tab and add riched20 and usp10 to 'native, builtin'. This implies that Wine should first try to load the native Windows libraries and if they are not available it should try the built-in libraries.



Wine configuration – library

> "We can use the native Windows libraries to make sure that applications have maximum compatibility"

05 Running Microsoft Office

Microsoft Office should now be available in the Ubuntu program menu. You can access it by going to Application> Programs>Microsoft Office (**Fig 2**).

Note

Most of the Office applications should work flawlessly, with the exception of Microsoft Outlook, Microsoft Publisher and Microsoft Groove 2007.

Building Windows applications using Winelib

Wine is really very helpful if you want to become Microsoft-free. Even a third-party Windows application developer is at the mercy of MS these days. For instance, Windows Multimedia application developers rely on DirectX for the simplest of tasks. DirectX is Microsoft's property, and nobody else has access to its source. Someday, Microsoft might create some distinctive features of DirectX which are not accessible to ISVs because they are undocumented. Windows Media Player would then be the best available multimedia player on the platform, because others were forced to rely on DirectX without having full control of it.

Welcome to the Linux arena, where nothing is impossible. Let's take the example of DirectX again. Wine has got is its own implementation of DirectX (built with the help of Transgaming Technologies). Wine implements common multimedia APIs, such as Direct3D, DirectInput, DirectSound, DirectShow and many others, by mapping them to powerful and open Linux equivalents like OpenGL, X11 and the ALSA/OSS API. Hence while porting your DirectX application to Linux you can use all these open technologies. Winelib can be used as a first step to get your application out of Windows and onto a Linux box. Then gradually you can take advantage of all that Linux has to offer.

Winelib is capable of running and compiling Windows applications on Linux without the need for Windows. Winelib is part of the Wine project. It is an open source implementation of the Win32 API built on top of Linux and the X Window System (often referred to as Xorg these days). Winelib shares 100 per cent of its code with Wine, and is capable of compiling both console and GUI applications. It is also capable of compiling library files like DLLs (dynamic-link libraries). Wine can handle your Windows source code in two ways...

> **"Winelib is capable of running and compiling Windows applications on Linux without the need for Windows"**

Providing binary-level compatibility

Binary-level compatibility is achieved with Wine itself. In this process, application source code is compiled on the Windows platform and then the binary file is taken to the Linux system and is run through Wine. When the application is run with binary compatibility, it can use all existing .dll files. This process is pretty straightforward, but is not able to unleash the full power of the Wine subsystem. This is exactly the same way we are running Microsoft Office 2007 using Wine.

Providing source compatibility

In this method the source code file is taken to the Linux box, where it is compiled against the Winelib libraries using GCC. This way, the application will also be able to catch up with UNIX API calls in order to leverage the full power of UNIX. Winelib ships with a tool called winemaker, which creates a GNU-standard autoconf-based makefile out of a VC++ project. Winemaker is a Perl script that does all the dirty work involved in converting the source code, making it UNIX specific, clearing up cases issues and a lot more.

Theoretically speaking, converting a VC++ application to a Linux application involves the following steps:

1. Copy the VC++ project to a directory on a Linux box.

2. Change to the above directory and issue the following commands...

```
$winemaker -lower-uppercase
$./configure -with-wine=/usr/wine/
$make
```

And you are done. You can now install or execute the built binary.

Hello World

Let's try Winelib on our favourite 'Hello World' Visual C++ application...

```
*--------hello.c ----------*/
#include <windows.h>
int main(int argc, char** argv) {
   MessageBox(NULL, "Hello Readers,
Welcome to Winelib.", "Hello",
MB_OK);
   return 0;
}
```

Compiling and executing

```
$ winegcc hello.c -o hello
$ ./hello.exe
```

Note

You will need GCC 4.5 to build applications with Winelib. To install GCC 4.5, enter the following command:

```
$ sudo apt-get install
gcc-4.5
```

Wine makes it relatively easy for your favourite Windows applications and games to run on Linux. If something is not working now, you may want to try the next version of Wine, as it is constantly being developed. If you are a Windows developer, Wine enables you to make your application natively available on the Linux platform. In future issues we will be looking into more Wine goodness.

Build packages for multiple platforms with the openSUSE Build Service

Build packages for software in the cloud, quickly and conveniently

Resources

osc openSUSE Build Service (OBS) tool http://download.opensuse.org/repositories/openSUSE:/Tools/ Source (Python), Git clone git://gitorious.org/opensuse/osc.git

osc-source_validator (Optional requirement) Client-side source validator tool. Search the Internet to fid the the RPM (for openSUSE, Mandriva, Fedora etc) and DEB (for Ubuntu, Debian etc) packages you're looking for

openSUSE Build Service Account Navigate to https://build.opensuse.org and click on the register link found at the top right of the page. Creating an account is free and it will be activated immediately

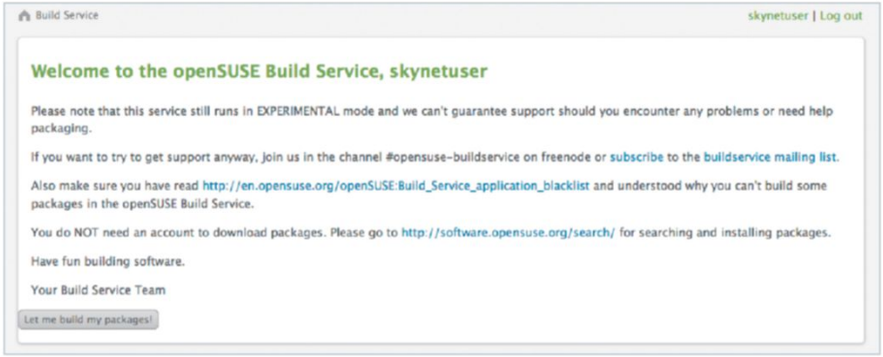


Fig 1 OBS welcome page

One of the most important aspects of software development is package management. Back in the old days, distributing a Linux package (or for that matter any other open source OS) would actually mean giving a tarball (tar.gz file) containing the source code of the package. A user would take this file, decompress it and compile it on the running distribution. This approach was taken for two reasons: (1) after all it is open source software, and what better way to distribute it than just distributing the source code?; (2) there was one package for multiple Linux distributions. For all its merits, tarball-based distribution has some serious issues. Compiling tarballs is difficult for average users. Also, you will need to have all the development packages and necessary compilers on your system and resolve all the dependencies manually. To get around this, binary distribution package formats such RPM and DEB were created. But even with these formats, the following problems were still there...

Building packages for multiple distributions: RPM/DEB packages created on one distribution may or may not work for another.

Dependency resolution: End users still have to install the dependent packages manually.

The openSUSE Build Service is a complete distribution development platform with support for multiple Linux distributions including openSUSE, Mandriva, Fedora, Red Hat, Ubuntu and Debian. **OBS provide developers with following advantages:**

1. Support for multiple Linux distributions.
2. Support for RPM and DEB packages.
3. Subversion-based integrated version control.
4. Automatic repository generation for all the supported packages.
5. Support for continuous builds and multiple users.
6. Automatic build-time dependency resolution for most packages.
7. Provides repository hosting solution.
8. Anyone can create a repository. No approval or fees are required.
9. Custom distribution and virtual appliance support.

Enough introduction, now it's time to use the openSUSE Build service first-hand...

Note

There are two interfaces available to use with the openSUSE Build Service. One is the web interface and the other a Python-based command-line interface called osc. For this tutorial we will be primarily using osc and will be referring to the web interface whenever required.

In this tutorial we will be building and publishing the application DOSBox using the openSUSE Build Service. DOSBox is an open source DOS emulator for BeOS, Linux, Mac OS X, OS/2 and Windows. It focuses primarily on running DOS games.

Advisor

Kunal Deo is a veteran open source developer. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris and Openmoko. He has written numerous articles on open source, Solaris and Linux related technologies for various technical magazines around the globe. In his free time he loves playing games on his Xbox 360 and PlayStation 3

01 Creating the home project

Every OBS user must have a home project to hold all your sub-projects and packages. When you log into the OBS homepage **https://build.opensuse.org/** for the first time, a welcome page (**Fig 1**) helps you to create your home project. Click on the 'Let me build my packages!' button. On the next screen, fill in the title and description and click Create Project to create your home project. Now your OBS account is ready.

Please keep the following things in mind while using the openSUSE Build Service:

1. The Build Service is still under development. It runs stable meanwhile, but there is no guarantee that no errors will happen at all.

2. Ask on the mailing list (**opensuse-buildservice@opensuse.org**) if you want to create projects outside of your home namespace (home:skynetuser:*).

3. Search for already existing packages before you create a duplicate. It is better to ask to join development instead of doing duplicate work.

02 Checking out the home project

Now you will need to check out the home project to your local directory. This is similar to how Subversion works. Create a local project directory. This will become your working directory for all OBS projects. Perform the following commands to check out the home project:

```
$ cd <project root dir>
$ osc checkout home:<OBS_username>
Your user account / password are not
configured yet.
```

You will be asked for them below, and they will be stored in /home/kunal/.oscrc for future use.

Creating osc configuration file /home/kunal/.oscrc...

```
Username: skynetuser
Password:
done
A    home:skynetuser
$ cd home:<OBS_username>
```

If this is the first time you are running the osc command, you will be prompted for your OBS username and password. Login details and other osc-related configuration is stored in ~/.oscrc file.

03 Setting up the default text editor

osc will try to open a text editor from time to time that will allow you to make changes, enter commit comment etc. While doing so it looks for an environment called EDITOR to look for an editor to open. You can set this variable to any editor you like. For example, here we are setting the editor environment to gedit.

```
$ export EDITOR=gedit
```

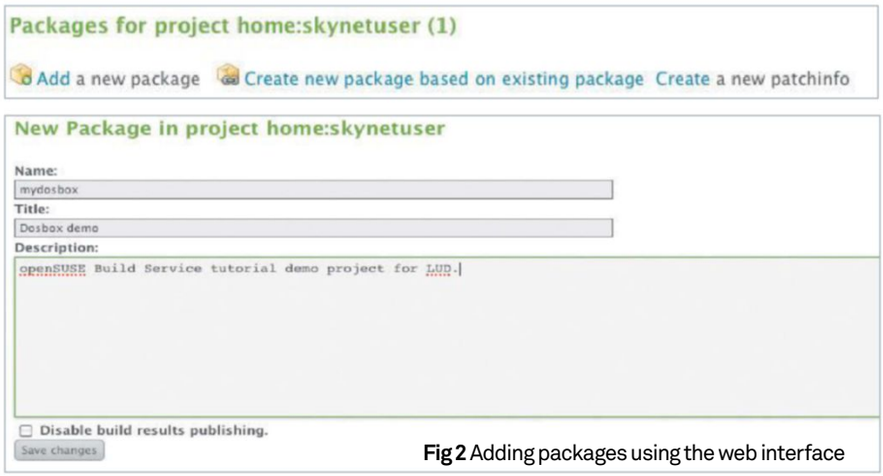


Fig 2 Adding packages using the web interface

04 Creating mydosbox package

The mydosbox package holds the files needed to build the DOSBox package. You can create multiple packages in your home project.

Perform the following command to create the mydosbox package.

```
$ osc meta pkg -e home:<OBS_
username> mydosbox
```

Add the meta information about the package in the open XML file. Save it and then close the editor window.

@code: Meta info XML file

```
<package name="mydosbox">
<title>Dosbox demo</title> <!--
Title of package -->
  <description>
openSUSE Build Service tutorial demo
project for LUD.
  </description>

<person role="maintainer"
userid="kunaldeo"/>
<person role="bugowner"
userid="kunaldeo"/>
<url>http://www.dosbox.com/</url>
</package>
```

In case you are facing any issues in creating the package, you can try creating the new package using the web interface. Go to Build Service>My Projects>home:skynetuser>Packages>Add a new package. Enter the meta information and click Save Changes (**Fig 2**).

Now update the working directory with the newly created package:

```
$ osc up
checking out new package mydosbox
A    mydosbox
At revision None.
```

This command will create a directory named mydosbox in the current directory.

05 Preparing the files for the package

Typically a package consists of the following files (for an RPM-based package)...

source.bz2 (or source.tar.gz/source.tgz): This is the source tarball that is available from the project site. In the case of DOSBox, the source tarball can be downloaded from **www.dosbox.com/download.php?main=1**.

package.changes (optional): This file holds the changelog for a given package. Changelog files are often present in the source file. You can copy the information from the same file to create the package.changes file. In this case we are copying the changelog file from the source tarball to dosbox.changes. If you have made any specific changes to the package, you append those changes in the dosbox.changes file.

@code: changes file for mydosbox, dosbox.changes

```
-------------------------------------
Wed May 19 11:30:33 UTC 2010 -
kunaldeo@gmail.com - new version
0.74
* Several small game specific fixes/
hacks/support. (Offensive, Roadhog,
GTA installer, Kingdom O' Magic
soundcard detection, Pirate booter,
Armored Fist installer)
* Add the S3-specific 640x480 256
color mode. (fixes regression in
"Wooden Ships and Iron Men" and "I
Have No Mouth And I Must Scream")
Fix a stack overflow that could
crash DOSBox.
```

package-patch.diff (optional): This file contains the patches that you are applying to the upstream project to make it work with the

distribution you are building for. This patch will be applied to source code during build time.

@code: patch file for mydosbox, dosbox-0.71-manpage.diff

```
Index: dosbox-0.71/docs/dosbox.1
===================================
 --- dosbox-0.71.orig/docs/dosbox.1
+++ dosbox-0.71/docs/dosbox.1
@@ -313,7 +313,7 @@ games (or
earlier). Also note that "prot
 Not all DOS programs work properly.
 .BR dosbox " will exit without
warning if an error occurred."
 .SH "SEE ALSO"
-The README in /usr/share/doc/dosbox
+The README in /usr/share/doc/
packages/dosbox
 .SH AUTHOR
This manual page was written by
Peter Veenstra <H.P.Veenstra@
student.rug.nl> and James Oakley
<jfunk@funktronics.ca>, for the
Debian system (but may be used by
others).
```

Note

It is recommended not to change the upstream source code directly, but rather have the changes in separate patch files. This helps the maintainer to understand the portion you have modified. Having patch sets also helps you send your changes back to the upstream project.

package.desktop (optional): The .desktop file represents the desktop entry file for the package. A desktop entry file describes how a particular program is to be launched, how it appears in menus, its desktop icon etc. This is a common specification adapted by all major desktop environments, including KDE and GNOME.

@code: desktop entry file for mydosbox, dosbox.desktop

```
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=DOSBox
GenericName=DOS Emulator
Comment=DOS emulator well-suited for
playing games
Exec=dosbox
Icon=dosbox
```

icon.png (optional): This is the icon file described by the desktop entry file.

```
@self: insert excerpt of the file
```

package.spec: .spec represents the RPM build spec file. The spec file is very important for a package as it drives the complete build process on the openSUSE Build Service. It governs how the source is configured, what patches are applied, what files will be installed, where they'll be installed, and what system-level activity needs to take place before and after a package is installed.

@code: dosbox.spec

```
# spec file for package dosbox (Version 0.74)
Name:           dosbox
BuildRequires:  Mesa-devel SDL-devel
gcc-c++ libpng-devel update-desktop-
files
BuildRequires:  SDL_sound-devel
BuildRequires:  SDL_net-devel
BuildRequires:  SDL_net
Url:            http://dosbox.
sourceforge.net/
License:        GPLv2+
Group:          System/Emulators/PC
Version:        0.74
Release:        3.8
Summary:        DOS Emulator Well-
Suited for Playing Games
# upstream URL for the source file
Source:         http://switch.
dl.sourceforge.net/sourceforge/
dosbox/dosbox-%{version}.tar.gz
# additional source files
Source1:        dosbox.desktop
Source2:        dosbox.png
# Name of the patch file
Patch:          dosbox-0.71-manpage.
diff
# Path for the build root folder
BuildRoot:      %{_tmppath}/%{name}-
%{version}-build
%description
dosbox is a DOS emulator that,
thanks to its good graphics and
sound emulation, is exceptionally
well-suited for playing games.
dosbox features a built-in DOS
operating system and transparent
access to the Linux file system and
is therefore very easy to use.
Authors:
--------
Sjoerd v.d. Berg <harekiet@zophar.net>
Peter Veenstra <qbix79@users.
sourceforge.net>
#Pre-build steps that include source
preparation, setup and patching
%prep
%setup -q
%patch -p1
#build step. Here the build scripts from the
source code are executed. This also includes
appropriate configure options. Configure
script options further configure many core
aspects of the package, including package
configuration, installation directory, info
pages directory, man page directory etc.
Compiler flags are also set up here. Do keep
in mind that additional compiler flags may
be added depending upon the distribution
you are building for. Lastly, we are executing
'make' to compile the package.
%build
autoreconf -f -i
CFLAGS="%{optflags}" \
CXXFLAGS="%{optflags} -fno-strict-
aliasing" \
   ./configure \
        --prefix=%{_prefix} \
        --infodir=%{_infodir} \
        --mandir=%{_mandir}
make %{?jobs:-j%jobs}
#Installation process. Installation is relative
to the --prefix option provided in the build
phase using DESTDIR as the root directory.
%install
make DESTDIR=%{buildroot} install
# We copy the document ourself
rm -rf %{buildroot}%{_datadir}/doc/
dosbox
# Updating the desktop items
%suse_update_desktop_file -i %name
Emulator
install -d -m 755 %{buildroot}%{_
datadir}/pixmaps
install -m 644 %{SOURCE2}
%{buildroot}%{_datadir}/pixmaps/
dosbox.png
# Clean up code
%clean
rm -rf %{buildroot}
# List of files that need to be included with the
RPM package. If your RPM package includes
files not owned by the RPM package, it may be
rejected during build time.
%files
%defattr(-,root,root)
%doc AUTHORS COPYING ChangeLog NEWS
README THANKS
%{_bindir}/dosbox
%{_mandir}/man?/*
%{_datadir}/applications/*
%{_datadir}/pixmaps/*
# Changelog for the RPM package. This could
be same as the .changes file.
%changelog
- new version 0.74
* Several small game specific fixes/
hacks/support. (Offensive, Roadhog,
GTA installer, Kingdom O' Magic
```

```
soundcard detection, Pirate booter,
Armored  Fist installer)

* Add the S3-specific 640x480 256
color mode. (fixes regression in
“Wooden Ships and Iron Men” and “I
Have No Mouth And I Must Scream”)
```

06 Adding files to package

Copy all the necessary files – such as the source tarball, spec file etc – to the local package directory.

Perform the following command to add the files to the package:

```
$ osc add *
A     dosbox-0.71-manpage.diff
A     dosbox-0.74.tar.gz
A     dosbox.changes
A     dosbox.desktop
A     dosbox.png
A     dosbox.spec
```

Commit the changes to the repository:

```
$ osc commit
Sending     dosbox-0.71-manpage.diff
Sending     dosbox.changes
Sending     dosbox.desktop
Sending     dosbox.png
Sending     dosbox.spec
Sending     dosbox-0.74.tar.gz
Transmitting file data ......
Committed revision 1.
```

osc will open the editor for the commit comments. Type in the comment and close the editor. osc will now start transmitting your files to the server.

07 Adding build targets and repository

Although we have everything ready on the openSUSE Build Service, there is one important thing still missing. We have not told the OBS which target to build for. The build target here means the distributions we will be targeting our packages for. You can build for any repository you want, including the ones you have created. But for this example let’s build our dosbox package for openSUSE Factory and openSUSE 11.3, on xi586 and x86_64 CPU architectures.

Build targets are added to the project metadata. Let’s do that now:

```
$ osc meta prj -e home:<OBS_
username>
```

@code: Project Metadata File

```
<project name=”home:skynetuser”>
  <title>kunaldeo’s Home Project</
title>
  <description>My home project.</
description>
  <person role=”maintainer”
userid=”kunaldeo”/>
  <person role=”bugowner”
userid=”kunaldeo”/>
  <repository name=”openSUSE_11.3”>
    <path repository=”standard”
project=”openSUSE:11.3”/>
    <arch>i586</arch>
    <arch>x86_64</arch>
  </repository>
  <repository name=”openSUSE_
Factory”>
    <path repository=”standard”
project=”openSUSE:Factory”/>
    <arch>x86_64</arch>
    <arch>i586</arch>
  </repository>
</project>
```

Lines in bold need to be added. To obtain a list of available repositories, use the following command:

```
$ osc ls
```

Or you can use the web interface to add the popular repositories, using Projects>Home Project>Repositories and clicking ‘Add repositories’ (**Fig 3**).

Once the build targets are ready, any commit or file changes will trigger the build process.

You can also manually trigger a rebuild if you need to:

```
$ osc rebuildpac <project> <package>
[<repo> [<arch>]]
```

08 Watching the build process and project repositories

You can watch the build log using the following command (this command needs to be run from the package directory):

```
$ osc buildlog <platform> <arch>
```

You can get the build results using the following command:

```
$ osc results
```

If it’s successful, packages will be published to your own repositories. **To show the URLs of .repo files which are package sources for Yum/YaST/smart, use the following command:**

```
osc repourls [dir]
```

You can distribute this repourls to enable others to download packages that you have just built.

Running your own copy of the openSUSE Build Service

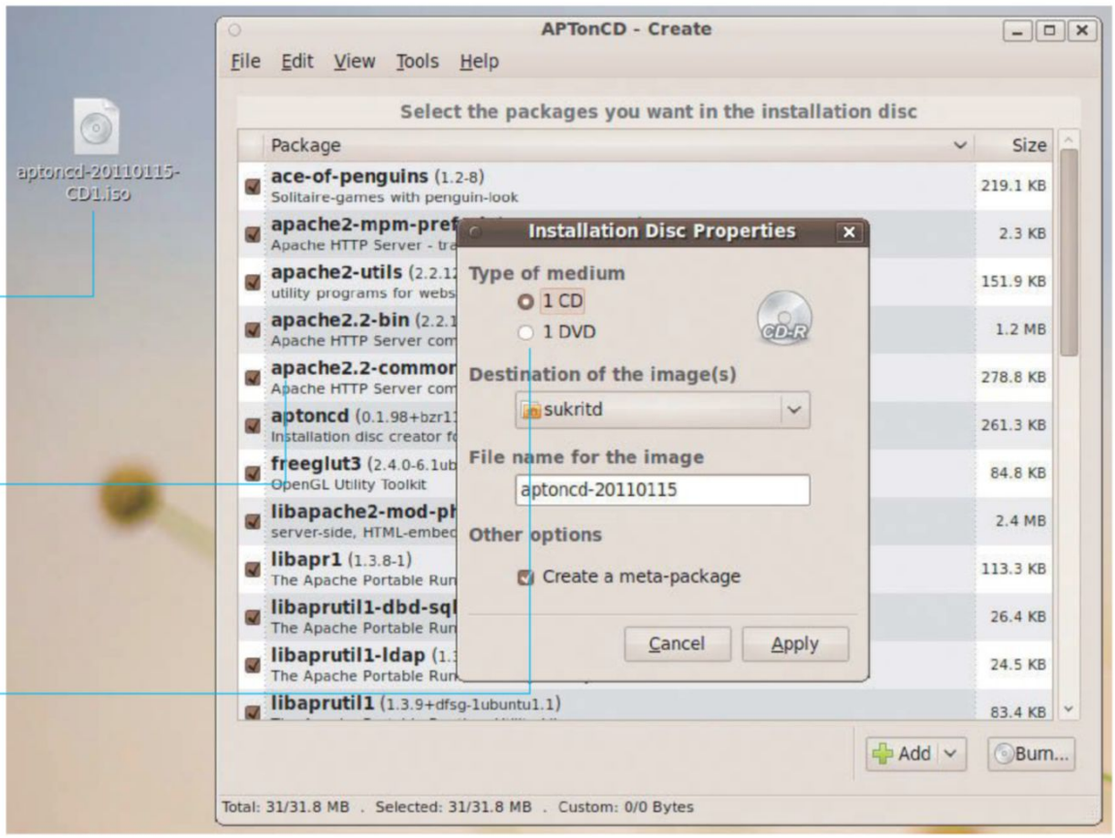
The beauty of the openSUSE Build Service is that the whole framework itself is also open source. This essentially means that if you want, you can install and use your own copy of the openSUSE Build Service inside your private network. Source code for the openSUSE Build Service is hosted on Gitorious at **http://git.gitorious.org/opensuse/build-service.git**.

You can also use OBS virtual appliances to get started with your own copy of the openSUSE Build Service in no time. These appliances can be downloaded from **http://download.opensuse.org/repositories/openSUSE:/Tools/images/iso/**.

Fig 3 Adding repositories using the web interface

Conclusion

The openSUSE Build Service makes it very easy to build packages for multiple distributions from a single source. This is one great step in making Linux free from the fragmentation problem. So next time you build a software package, do not just build it for your favourite distribution – build it for all major distributions.



The app saves a copy of the backup as an ISO image which you can store on a thumb drive if you prefer

The list of packages that are to be burned onto the backup disc

You can choose if you want to create CDs or DVDs containing the backup

Create a portable backup of your packages with APTonCD

APTonCD is a simple application that helps you take a backup of all the applications you have installed in Linux. Here's how it works...

Resources

APTonCD APTonCD is a tool with a graphical interface which allows you to create CDs or DVDs with the packages you've downloaded via apt-get
http://aptoncd.sourceforge.net

Has your Linux install ever given in after you performed several tweaks and package installs to create the perfect system for your work and entertainment? Reinstalling the machine from scratch can be quite a frustrating feeling. Well, APTonCD is a project that will come to your rescue in such scenarios. APTonCD allows you to take a backup of all the applications you have installed using tools such as apt-get and Aptitude, and burn it to CDs or DVDs. You can use these discs to then install further systems or to recover all your installed apps when you need to reinstall your computer.

01 Getting APTonCD

The APTonCD project can be found at http://aptoncd.sourceforge.net/. Here you will find a lot of useful documentation and links to help you with the project. You will also find the binaries required for installation here. Go to the Download section of the website. You will see two download options – one for a deb and another for a tar.gz. Pick the Debian binary or the source file, depending upon your requirements. Note that this project supports only Debian and Ubuntu Linux.

The Download section of the APTonCD project website

> **Allows you to take a backup of all the applications you have installed using tools such as apt-get and Aptitude**

02 Installing binary or source

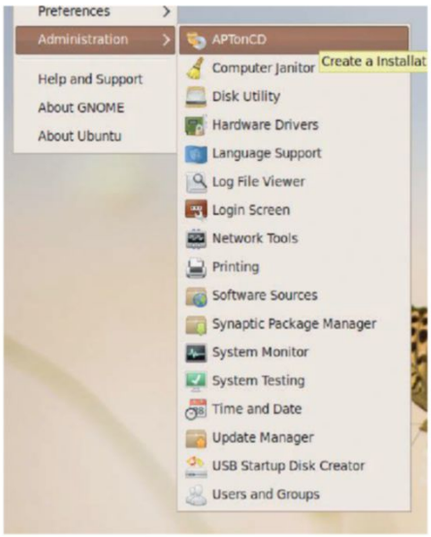
To install the Debian binary file, double-click on it and follow the steps. To install APTonCD from source, extract the tar.gz file, get into the extracted directory, then execute the command '# sudo make install'.

03 Installing with package manager

The easiest way to install APTonCD on Ubuntu Linux is to either execute the command '# sudo apt-get install aptoncd' or to search for it in Add/Remove Software Application and hit the Install button (**Fig 1**).

04 Launch the app

After the installation is complete, you should be able to launch APTonCD from the system menu. Go to System>Administration >APTonCD. When the app launches, you will immediately see how simple an application it is. Despite its simplicity it is a very effective program. The developers have been smart enough to reduce the options available to users to a minimum, making the app easier to use.



APTonCD does not go into the Applications menu like most other applications

05 Getting started

The first launch pane of APTonCD will ask you whether you want to do create a backup disc or restore from a backup. Let's begin with the creation of the backup. To do this, click on the Create button.

The first screen offers two options: to create a backup disc or to restore from a backup

06 Select packages

Now you will be presented with a list of packages. This list comprises packages that you have installed on your system additional to those that would have come with your distribution. Note that this list only features packages that were installed using the apt-get based application such as Aptitude and Synaptic. The next order of business is to select the packages that you want backed up.

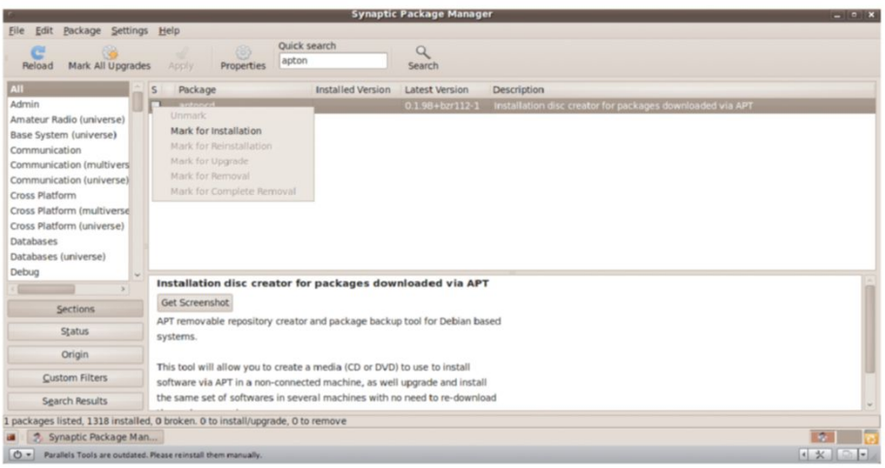


Fig 1 Installing with package manager The installation process using SynapticCMS

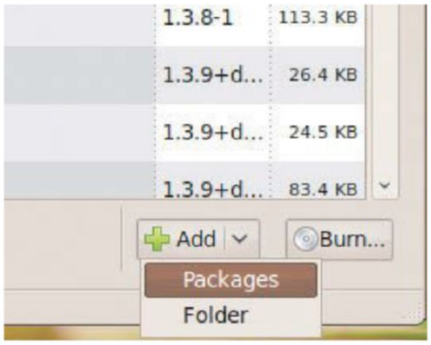
A list of all packages available for backup

07 How to select packages

If you are not short of space on your media (CDs of DVD), you can just burn them all (**Fig 2**). However, if you need to select some apps that you don't want to put onto your backup disc, we suggest you pick those applications that are updated frequently. For example, the Firefox web browser is something that is updated with reasonable frequency.

08 Additional packages

Once you have finished sorting the packages that are installed on your computer, you can customise things a bit more. There's a button at the bottom of the pane, titled 'Add'. If you click on this button, you are offered two options: to add packages or folders.



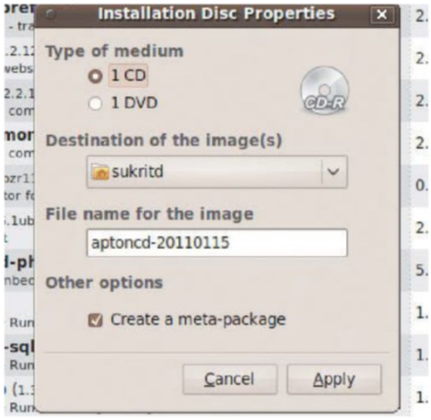
The Add Packages option menu

09 Customising further

Using this option, you can add more Debian packages or folders containing packages. It might not always be the best thing to do as you might face compatibility issues. You can also drag and drop Debian packages right into the package-selection pane to add them to the list.

10 The disc image

After you have finalised the selection of the packages with which you want to create a local repository disc, you can click on the 'Burn' button. This will take you to the window where you choose what kind of media you would like to burn these files onto – CD or DVD – and the location where to store this ISO image. APTonCD will then offer to burn the CD or DVD with the default burning application, or you can use your favourite app such as K3B to burn the ISO image onto a disc.

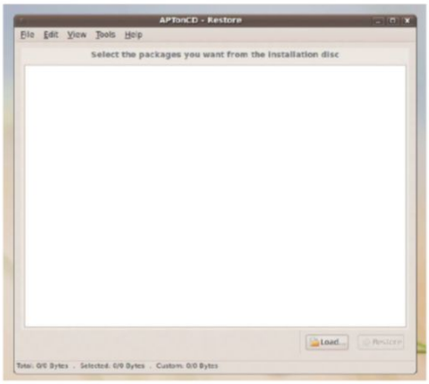


In a matter of a few minutes, your APT package backup should be ready and waiting for you

11 The restoration process

Assuming that you have managed to create the Debian binary backup disc correctly, the next major feature of APTonCD is the restoration process. Let's hope you don't have to use this for all the wrong reasons, but here's how it works. APTonCD creates an image-based repository of all the Debian packages that it found on your system and allows you to add them to your system.

Fig 2 How to select packages You can view the version of each package by enabling the option from the View menu



Until you load packages into the Restore pane, it will lie empty

12 APTonCD Restore option

It would have been nice if you could have restored the entire set of packages you created as part of your backup with just one click. However, as of now this is not quite possible. This feature needs a major overhaul. What you can do though is to restore the APTonCD backup CDs or DVDs you created to your installation as a repository. To do that, launch APTonCD from the System menu and hit the Restore option. When asked for the media, direct it to the CD, DVD or ISO image you made.

APTonCD is an excellent piece of software, but it currently lacks support. As members of the open source software community, whether as developers or users, it would be great if you could start using this project and encouraging the APTonCD project to complete the restoration feature so that more people can use it.

Advisor

Sukrit Dhandhania has

spent several years working professionally, implementing different open source tools for companies. During this time he has evaluated, set up and maintained various open source tool for these firms

Load a CD or a DVD containing the APT packages' backup

13 A better way to restore

As mentioned in the previous step, the Restore feature in APTonCD needs some work. As it stands, it is just about functional. There is a rather simple workaround to it, though. Pop in the CD or DVD, or mount the ISO image you made with the package backup. APTonCD should mount the image for you; if not, to mount the image, execute the command '# sudo mount -o loop /path/to/image/imagename.iso /mount/to/folder/' (**Fig 3**). You should now be able to view the files contained in the image.

14 The good old command line

Launch a Terminal window and change into the directory where you mounted the disc or image. Execute the step '# sudo dpkg -i *deb'. This step might take a few minutes, or longer, depending upon the speed of your computer and the number of packages contained in your backup disc (**Fig 4**).

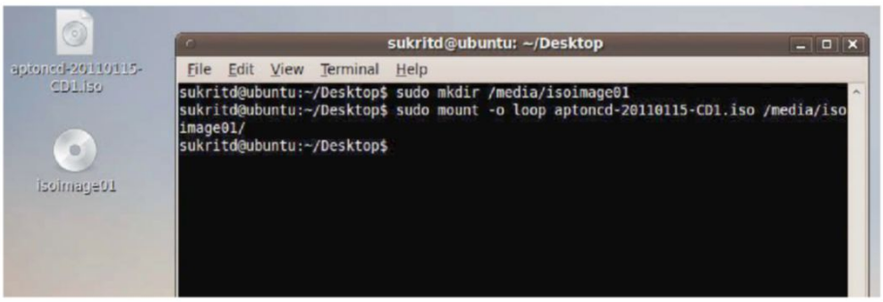
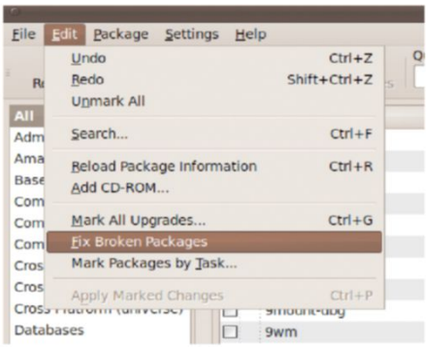


Fig 3 A better way to restore An example of how to mount the ISO image manually

15 Fix the dependencies

If you have a working internet connection on your computer, launch the Synaptic package manager from the System menu and fix any existing dependencies that may have occurred during the installation of these packages. In an ideal scenario you should not have any issues, but there's no harm in checking.



Use the option in Synaptic to fix any dependencies that might have occurred during the mass installation

16 Check your apps

Once you are done, click on the Applications menu and check to make sure that all the packages you have backed up have been installed correctly. Voila, you have now completed the entire cycle of backing up and reinstalling the packages on your computer.

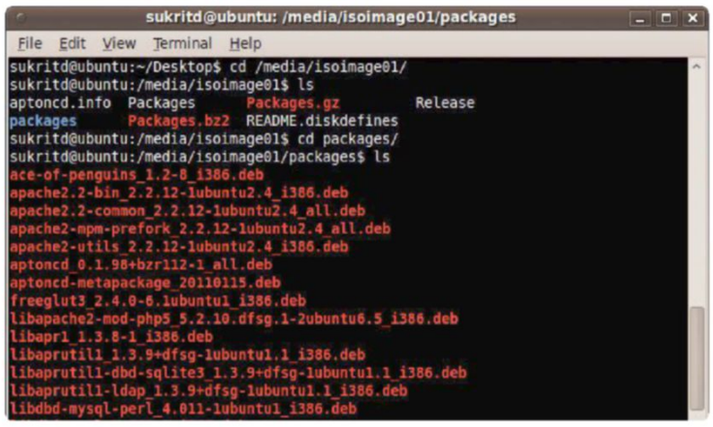


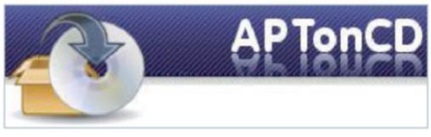
Fig 4 The good old command line As you can see, all the packages from the APT cache directory were copied here

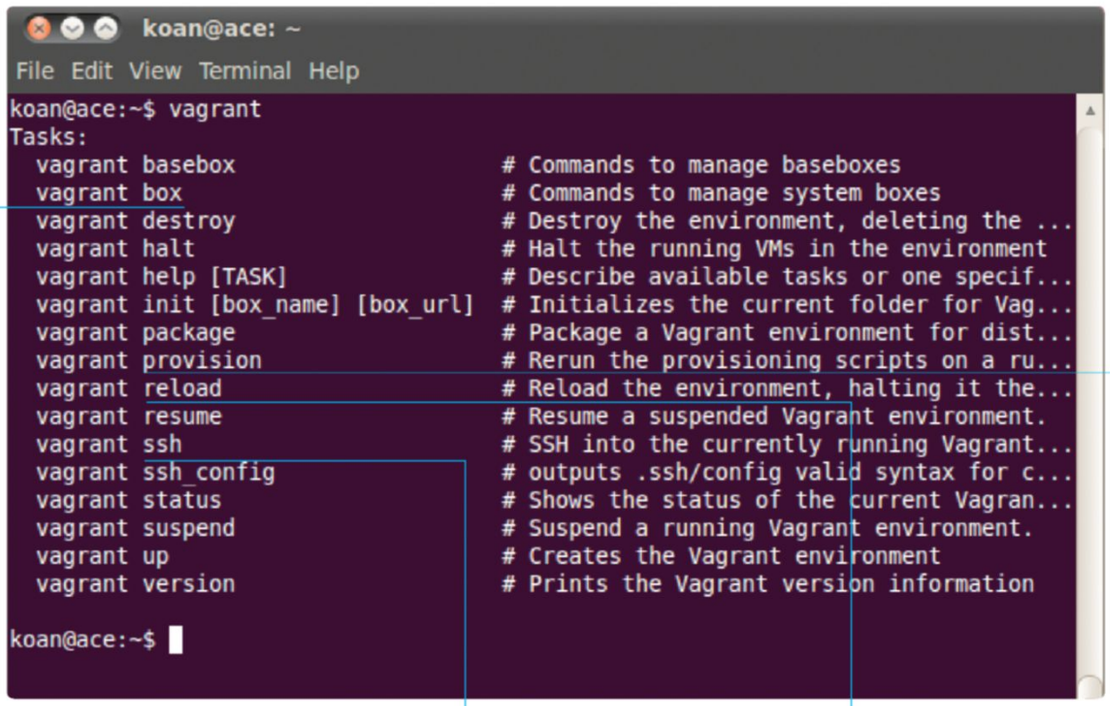
17 Behind the scenes (how it works)

What goes on behind the curtain when you ask APTonCD to back up your packages is that it picks up the list of software binary packages available at /var/cache/apt/archives. These are packages installed by apt-get based systems such as Aptitude and Synaptic. The process involved is quite simple, but what is really nice is that the APTonCD team has made it accessible to not very advanced users by throwing in a simple graphical window to it.

During the Restore process, what happens is that all the packages that are on the disc or ISO image are copied back to the /var/cache/apt/archives directory. It would have been nice if the app could also install them for you, but we hope it will soon. In the meantime you need to install the packages either using the Synaptic package manager or using the command line, as shown earlier.

APTonCD can be a lifesaver, especially for people without a proper internet connection. It can also be of great use to system administrators who need to install the same set of packages on several computers. While being quite effective, it also gains a lot of points in being fairly easy to use. However, while the process of backing up is as simple as could be, the software needs to seriously work on the restoration process. Where it stands today, APTonCD loses out on a lot of potential users by not fixing this feature.





A central concept in Vagrant is the 'box', which is a barebones operating system used as a template for virtual machines

Log into your virtual machine using SSH

You can boot, shut down, reload and destroy a virtual machine with just one command

Coupled with a configuration management system like Puppet, Vagrant is able to customise your virtual machines

Virtual machines made easy

Using Vagrant, you can automatically generate virtual machines for Oracle's VirtualBox, meaning there's no need to battle through the installer windows any more

If you have a small development company and you want to get new employees up and running quickly with a development virtual machine, you can save a lot of time by automating the creation of virtual machines. But also, if you're just a solo developer it could be useful to get a complete development or test environment up and running in no time. That's where Vagrant comes in, a tool for automated virtual machine creation for Oracle's VirtualBox. It provides easy-to-configure, lightweight, reproducible and portable virtual machines targeted at development environments.

Apart from installing a virtual machine, you can configure Vagrant to forward ports to your host machine, to configure shared folders, and so on. You can also tie in a configuration management system like Puppet (see **Linux User & Developer** issue 89) to Vagrant to specify a complex configuration for your virtual machine. Moreover, it's possible to package a virtual machine in a distributable box. Vagrant makes all of this really easy. Even rebuilding a complete environment from scratch or tearing down the environment when you're done is possible with a single command, and it happens in a flash.

Advisor

Koen Vervloesem has been writing about free and open source software, and IT in general, since 2000. He has master's degrees in computer science and philosophy and can be reached at koen@vervloesem.eu

Resources

VirtualBox 4.0 http://www.virtualbox.org/

Vagrant http://vagrantup.com/

VeeWee https://github.com/jedi4ever/veewee

01 Install VirtualBox

Vagrant depends on VirtualBox version 4.0.x to create its virtual environments. So, first, install this version on your computer or update your older VirtualBox to the latest release. The Linux downloads page on **www.virtualbox.org** gives instructions for most mainstream Linux distributions. We installed VirtualBox 4 from the VirtualBox repository on an Ubuntu 10.04 host.

02 Install Ruby

Vagrant is written in Ruby, so you have to install some packages for this programming language. On our Ubuntu host, we run 'sudo apt-get install ruby-dev ruby ri rdoc irb libreadline-ruby libruby libopenssl-ruby wget'. This installs Ruby, Ruby development headers, Ruby documentation, the Ruby debugger, some Ruby libraries and Wget (which we'll use in the next step to install RubyGems).

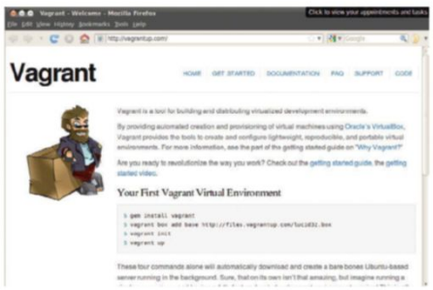
03 Install RubyGems

We also have to install RubyGems, a package manager for Ruby modules. **However, the version in the Ubuntu repositories is too old, so we install RubyGems from source:**

```
$ cd ~
$ wget http://production.
cf.rubygems.org/rubygems/rubygems-
1.6.1.tgz
$ tar xvzf rubygems-1.6.1.tgz
$ cd rubygems-1.6.1
$ sudo ruby setup.rb
$ sudo ln -s /usr/bin/gem1.8 /usr/
bin/gem
```

04 Install Vagrant

Now we can easily install Vagrant as a RubyGems module with 'sudo gem install vagrant'. This installs again a lot of dependencies, all of them RubyGems modules used by Vagrant, such as SSH and SCP bindings for Ruby and an interface for Ruby code to communicate with VirtualBox.

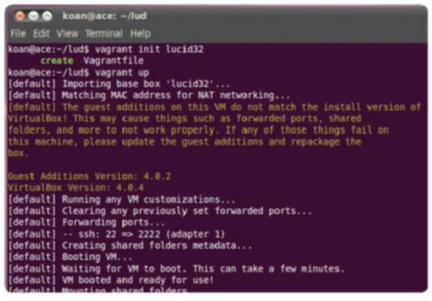


05 First Vagrant virtual environment

We're ready now to set up a virtual machine with Vagrant. **Just create and enter a new directory (this will be the directory for our Vagrant project for the rest of this article) and use these three commands:**

```
$ vagrant box add lucid32 http://
files.vagrantup.com/lucid32.box
$ vagrant init lucid32
$ vagrant up
```

This downloads an Ubuntu 10.04 template, initialises a new virtual machine based on it and starts it up.

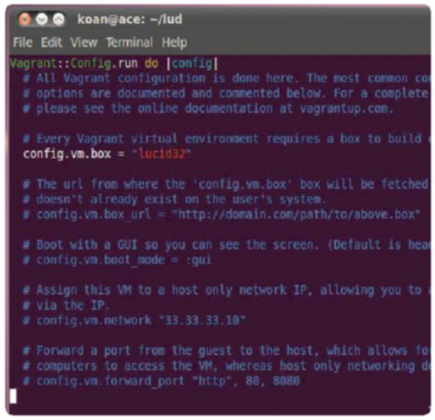


06 The vagrant binary

The vagrant binary, which is a command-line interface for Vagrant, has many subcommands. In the previous step, we already used the subcommands box, init and up, and if you want to discover all the supported subcommands, just run the vagrant command without any options, which will list them for you.

07 The Vagrantfile

When you entered the 'vagrant init' command, Vagrant created a file named Vagrantfile in the current directory. A Vagrantfile is to Vagrant as a makefile is to Make: it specifies the behaviour of the virtual machine that Vagrant creates. A Vagrantfile is written in Ruby code, but almost all commands in the file are just variable assignments. For example, the line 'config.vm.box = "lucid32" ' specifies the box the virtual machine is based on.



08 Exchanging files

By default, Vagrant shares the directory where you have initialised the virtual machine, and in the virtual machine this directory is available under the path /vagrant. This means that your virtual machine has read-write access to its Vagrantfile, but also to any other file you add to the same directory. **An example:**

```
$ echo "<h1>Hello world from a
Vagrant VM</h1>" > index.html
```

09 SSH

When you run 'vagrant ssh' in your project directory, Vagrant will get you an SSH login for your virtual machine (**Fig 1**, overleaf). Your username is 'vagrant', but SSH is configured not to ask for any password. You can also run commands with sudo without being asked for a password. Remember, we shared a folder, so take a look at this folder with 'ls /vagrant'. You should see a Vagrantfile and index.html.

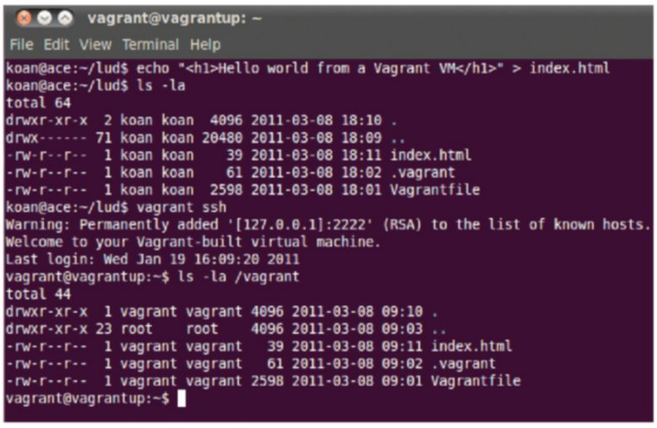


Fig 1 SSH Vagrant creates an SSH login for your virtual machine

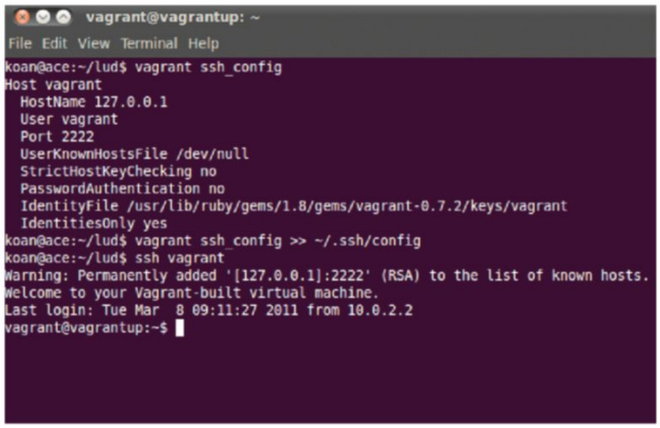


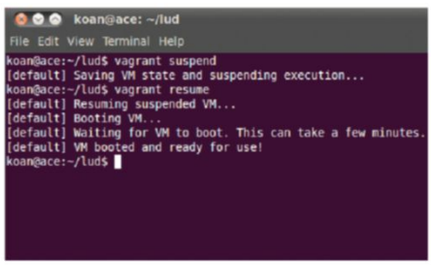
Fig 2 SSH configuration Use the 'vagrant ssh_config' command

10 SSH configuration

You can also use just SSH directly (or use SCP to copy files to/from the VM), but this requires you to know some SSH configuration. Luckily, Vagrant is very helpful here: just use the 'vagrant ssh_config' command, which outputs the configuration lines you have to put into your ~/.ssh/config to successfully connect to the virtual machine (**Fig 2**).

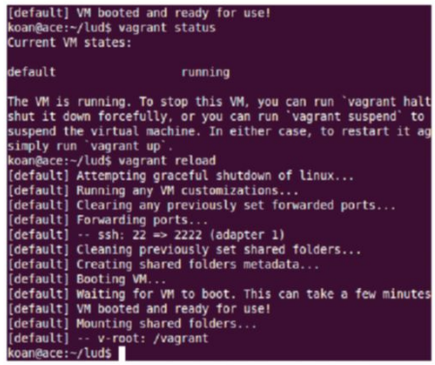
11 Suspend the virtual machine

With the 'vagrant suspend' command, Vagrant will save the current running state of your virtual machine and then stop it. When you want to resume working again with the virtual machine at a later time, you just enter a 'vagrant resume command' and 10 to 15 seconds later you're up and running again.



12 Halt the virtual machine

If you really want to shut down the virtual machine, you can do this with the 'vagrant halt' command, which issues the halt command in the virtual machine. You can later boot the machine again with the 'vagrant up' command that we used in the beginning. You can also reboot a running virtual machine with one command, 'vagrant reload'. This is especially useful if you have changed the VM's Vagrantfile and you want to apply your changes.



13 Destroy the virtual machine

The 'vagrant destroy' command deletes the virtual machine's image (so only do this if you don't need the data in the VM any more!). This could be useful if you've messed up your virtual environment and want to start with a clean slate, or simply if you don't need it any more. However, this doesn't delete the Vagrantfile, so you can always rebuild the environment from scratch with a 'vagrant up' command.

14 Customise with Puppet

Until now you still have a fairly standard Ubuntu system, but now comes the crux: you can tie in a configuration management system like Puppet or Chef to Vagrant so you can customise your virtual machines. You can use a Puppet server to configure your virtual machines (see our article in Linux User & Developer issue 89), but Vagrant also supports a standalone mode which doesn't require a Puppet server: just use 'config.vm.provision :puppet' in your Vagrantfile.

15 Puppet manifest

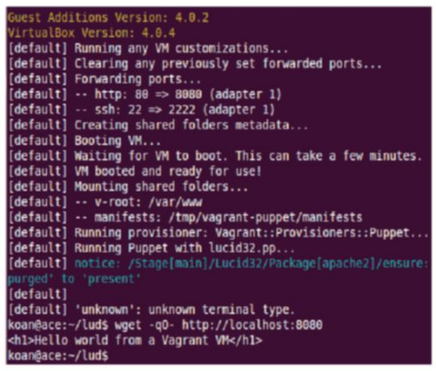
Now you specify the customisation of your virtual machine in a file called boxname.pp (in our example, lucid32.pp) in a directory 'manifests' inside our project's directory. This manifest file will contain the required Puppet configuration. For example, with the Puppet manifest shown in the image (**Fig 3**), we install an Apache 2 web server. After the configuration is complete, run 'vagrant reload' to reboot the virtual machine or run 'vagrant provision', which (re-)applies the Puppet configuration.

16 Port forwarding

If you want to forward port 80 on the virtual machine to port 8080 on your host machine, just specify this as follows in the VM's Vagrantfile:

```
config.vm.forward_port("http", 80, 8080)
```

The first argument "http" is just a name so you remember later what you need this port for. After you have restarted the virtual machine, you can point your web browser to http://localhost:8080/ to see your website.



17 Package your VM

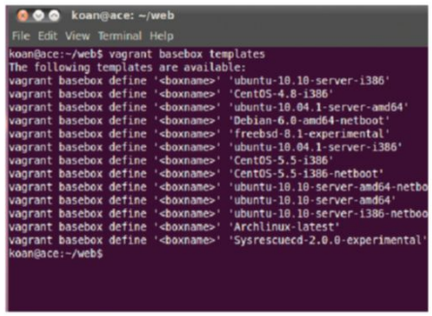
Now, if you want to share this customised virtual machine with others, you can package it, which makes a 'box' of it that others can use as a template for their own virtual machines. Create a Vagrantfile named Vagrantfile.pkg that specifies port forwarding for the services running on the box (eg the web server) and then package the environment with the command 'vagrant package --vagrantfile Vagrantfile.pkg'. The result is a file called package.box in the project's directory.

18 Distribute your box

Now you can distribute the .box file on your website or even on a local disk (or shared directory). Other users can download the box to their Vagrant installation and create a virtual machine based on it, just like you did in the beginning of this tutorial with the default lucid32 box. By the way, if you ever want to see which boxes are installed on your computer, run 'vagrant box list'. You can remove a box you don't need any more with 'vagrant box remove <boxname>'. Note that this doesn't touch any of the virtual machines based on this box.

19 VeeWee

If you really want to delve deeply into building your own Vagrant base boxes that are not based on the default Ubuntu Lucid box, there's an interesting tool that can help you: VeeWee. The latest release didn't work for us, but we installed version 0.1.11 successfully with 'sudo gem install veewee -v 0.1.11'. After this, the vagrant binary is extended with a basebox subcommand. You can list the available templates with the command 'vagrant basebox templates'.



20 Baseboxes

If you want to create a new basebox, you just have to run one of the commands that were shown in the output of the 'vagrant basebox templates' command. The only thing you have to do is change the boxname. **For example, if you want to create a CentOS basebox for Vagrant, run the following command:**

```
$ vagrant basebox define 'centos5.5' 'CentOS-5.5-i386-netboot'
```

This creates a directory – definitions/centos5.5 – in the current directory, with files that define the basebox. You can customise the default values, eg the memory and disk size, in definition.rb.

21 Create a box

Now you can build a box based on this basebox with the command 'vagrant basebox build centos5.5'. The command will ask you to download the ISO file. After the basebox has been built, you export the virtual machine to a .box file with 'vagrant basebox export centos5.5', and then you import this box file into Vagrant with 'vagrant box add centos5.5 centos5.5.box'. From now on, you can use this box as the base of your virtual machines with 'vagrant init centos5.5'.

> **It's very easy to build a virtual environment from scratch**

22 Some VM specifics

We end this tutorial with some tips. You can change the VM's name (as seen in VirtualBox) and the memory size. The latter is especially interesting if you need more horsepower in your VM, as the lucid32 basebox only uses 512MB by default. **An example configuration in the Vagrantfile:**

```
config.vm.customize do |vm|
  vm.memory_size = 1024
  vm.name = "Linux User & Developer VM"
end
```

23 NFS shared folders

We used VirtualBox shared folders, but it is well known that this feature has an awful performance if the folder has a large number of files. Luckily, Vagrant makes it really simple to use NFS: just add ':nfs => true' to the shared folder configuration in your Vagrantfile and use host-only networking, eg:

```
config.vm.share_folder("v-root", "/vagrant", ".", :nfs => true)
config.vm.network("33.33.33.10")
```

Of course, your host machine must have nfsd installed and running (in Ubuntu: 'sudo apt-get install nfs-kernel-server'), and your VM must have NFS support installed. Run 'vagrant up' or 'vagrant reload' after changing the Vagrantfile.

24 Disposable virtual machines

Vagrant gives you disposable virtual machines: if you ever want to test something, it's very easy now to just build a virtual environment from scratch, try something out and tear down the virtual machine. This is not only useful for developers, but also for system administrators and essentially for everyone who has to install some complex software from time to time. The next time you try out some new software, don't clutter your own desktop computer with a number of potentially conflicting dependencies, but fire up a VM with Vagrant and install the software there.

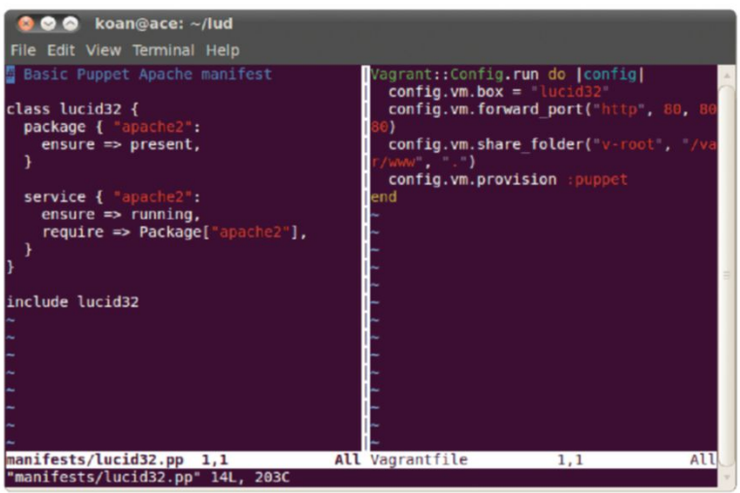
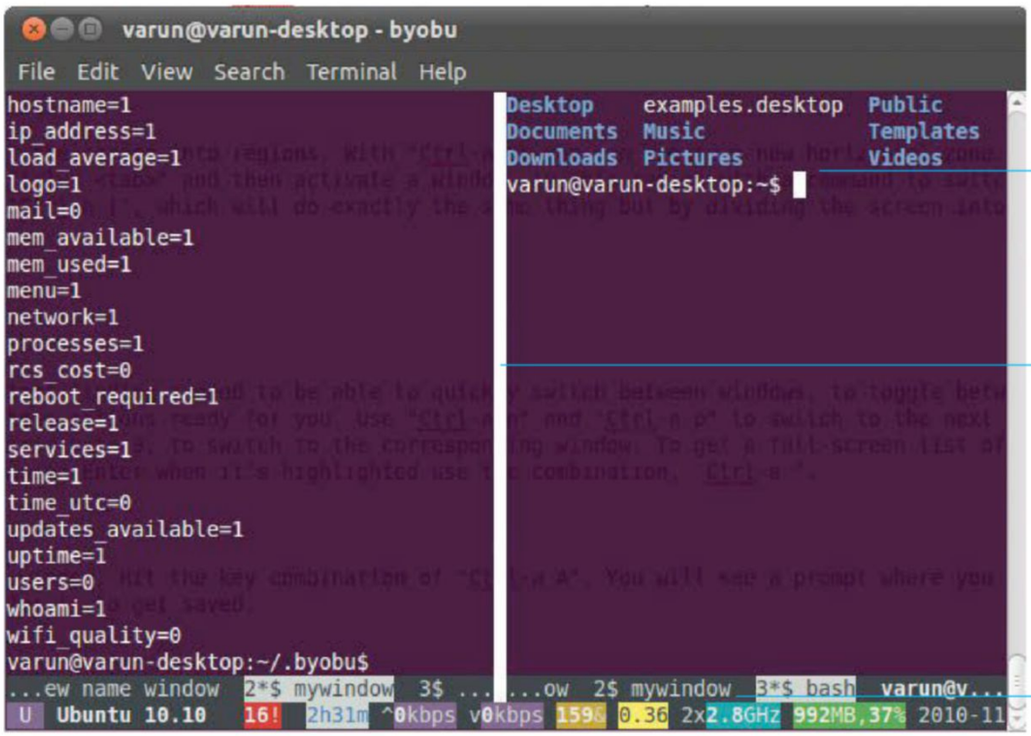


Fig 3 Puppet manifest
Specify the customisation of your virtual machine



Improve your remote SSH sessions with Byobu

Byobu is an enhancement for GNU Screen that features lots of powerful shortcuts and is incredibly easy to use. Here's how to improve your remote SSH experience the easy way...

Byobu is a Japanese term for decorative, multi-panel screens that serve as folding room dividers. Byobu is also the name of a project that enhances GNU Screen. If you have used or regularly use the GNU Screen utility, you might have an idea about just how much it can help enhance your remote SSH sessions. It gives you a lot more power and flexibility than the regular Terminal window would otherwise. The issue with Screen, though, is that it can be a bit awkward to get to grips with – the polar opposite of Byobu...

Advisor

Sukrit Dhandhania has spent several years working professionally, implementing several open source tools for companies. During this time he has evaluated, set up and maintained various open source tools for these firms

Resources

Byobu is an enhancement package for the GNU Screen application **https://launchpad.net/byobu**

01 Install Byobu on Meerkat

Getting Byobu is pretty simple, at least on Ubuntu Linux. If you are using the latest release of Ubuntu, which is Ubuntu 10.10 Maverick Meerkat or better, you should be able to find Byobu in the apt-get repositories. Launch the Synaptic Package Manager, search for 'byobu' and install the package (**Fig 1**).

02 Install on others

If you are using another distribution of Linux, check in its application repository to see if there is a binary. If you are using another version of Ubuntu Linux and you can't find the app in the apt-get repositories, you can use the following

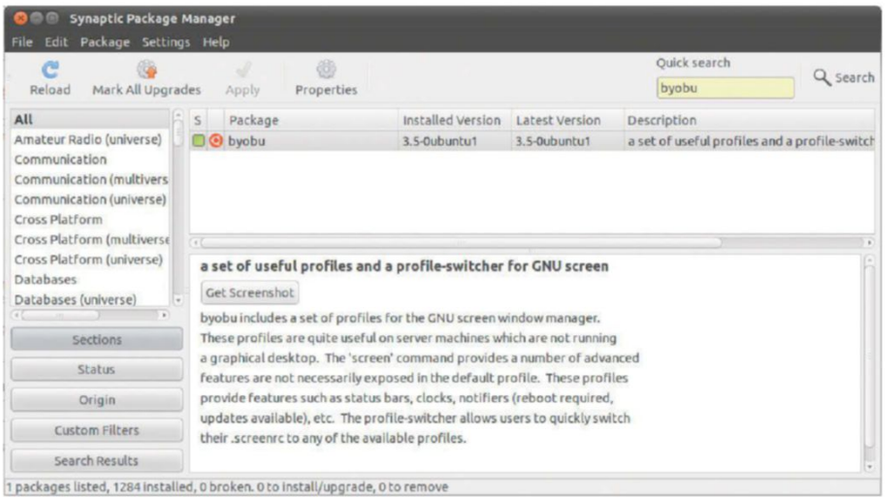


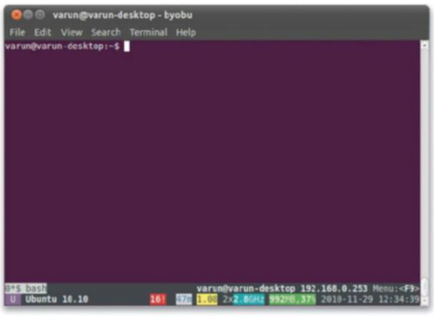
Fig 1 Install Byobu on Meerkat The Byobu installation process on Ubuntu Meerkat is very simple indeed

method to install Byobu. Note: Replace the term 'UBUNTU_VERSION' with the name of your release, such as jaunty, intrepid or hardy.

```
# sudo bash -c "echo 'deb http://
ppa.launchpad.net/byobu/ppa/ubuntu
UBUNTU_VERSION main' >> /etc/apt/
sources.list"
# sudo apt-key adv --keyserver
keyserver.ubuntu.com --recv-keys
F430BBA5
```

03 Test Byobu

Execute the command '# byobu' from the Terminal window. You should see the window change and look something like the screenshot below. However, if you received an error stating something like 'Error: Can't open slave tty /dev/pts/1 -- Permission denied', you will need to fix some permissions. Execute the command '# sudo chmod a+rw /dev/pts/1'. Change the 1 to the number from the error message.



What Byobu looks like at first launch, with its default settings

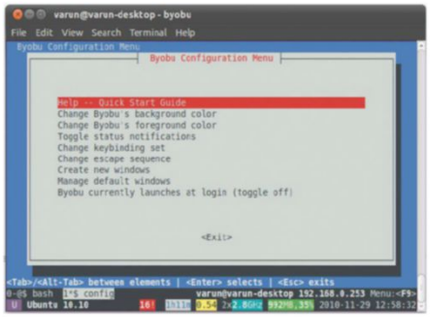
04 The Byobu Interface

You will note that there is a colourful strip of information being displayed at the bottom of the Terminal window. This is part of the enhancements that Byobu bring to the Terminal. By default it displays some basic information like the machine's name and IP address, along with a few statistics about the state of the machine, such as the amount of RAM and CPU being used. **Here's a more complete list of things that Byobu can display in the strip at the bottom of the Terminal screen.**

1. Screen windows list
2. Battery status
3. CPU count
4. CPU frequencies
5. Current date/time
6. Disk space
7. EC2 cost
8. Fan speed
9. Hostname
10. IP address
11. Load average
12. Mail count
13. Memory available/used
14. Network transfer speeds
15. Temperatures
1. Processes running (count)
2. Users logged-in (count)
3. Wi-Fi quality
4. Updates available

05 Customise Byobu

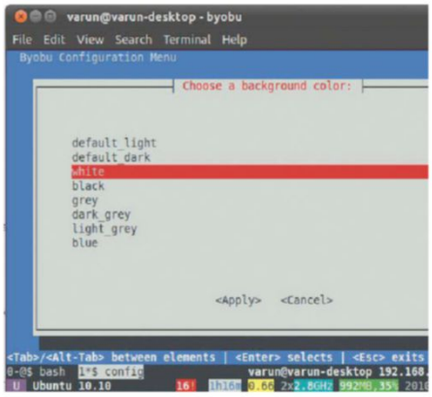
With so many more options to view information, it is easy to customise Byobu's configuration. Hit F9 to load the tool's configuration module. This configuration interface works as a textual menu. You can navigate it by using the up and down arrows. Hit the Tab button to move from one section to another. Use the spacebar to select or deselect an item. If you need help at any time, hit Enter on the 'Help -- Quick Start Guide' option.



The Byobu application menu

06 Change the look and feel

The first set of customisations we'll look at will enhance the look and feel of the Terminal. Byobu gives you a couple of options on this front. You can change the background and foreground colours being used in the Terminal. Enter the two sections for picking colours, choose the colour you like for each option and then hit the Apply option. If you don't like it, you can always return to the menu and change the colour.



Pick a colour to customise the look of your Terminal windows

07 Status notifications

We've found the status notification bar at the bottom of the screen to be immensely useful. The fourth option in the Byobu config menu is to configure what goes into this status notification bar. Hit Enter on this menu option to start. Now pick the info you want displayed constantly on

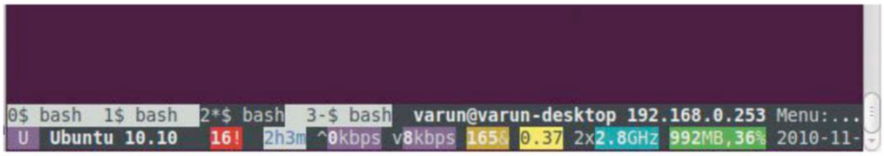


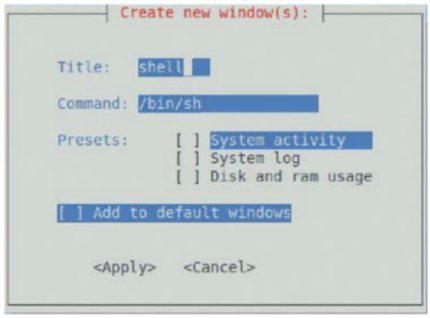
Fig 2 The new window You can see a list of windows in the status bar

your screen. Note that you do not want to go overboard with your selection as it could be quite distracting. Choose the Apply option when done.

■ Select the notifications you want to see in the status bar

08 New window options

If you are familiar with using GNU Screen, you might be aware of the powerful virtual Terminal interface that it provides you with. You can access this option in Byobu by selecting the 'Create new windows' option. Here you are presented with some options. You can pick which shell you want to use for the window, you can give it a name, you can choose from some presets, and you can add it to your list of default windows. Hit Apply to create a new window.



■ Launch a new window through the menu

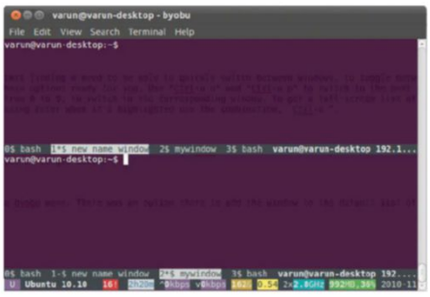
09 The new window

What happens when you hit Apply in the previous step is that you are immediately taken to the window that you just created. You can use the F3 key to move around between windows that you have created. You should be able to see a list of the windows in the byobu status notification bar as you create them (**Fig 2**).

10 Split the screen

Another very interesting feature makes it possible to split the screen into regions. With 'Ctrl-a S' you can create a new horizontal zone. This area will not contain anything at first, but you can move the focus there with 'Ctrl-a <tab>' and then activate a window in this region with a command to switch windows as 'Ctrl-a N'. If you prefer a vertical division of the windows, you can use 'Ctrl-a |', which will do exactly the same thing but by dividing the screen into two regions with a vertical dividing line.

> **"Another very interesting feature makes it possible to split the screen into regions – either vertically or horizontally"**



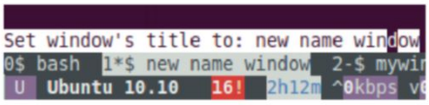
■ A horizontally split window with two different operations going on in parallel

11 Moving between windows

As you get comfortable with working with multiple virtual windows, you will start finding a need to be able to quickly switch between windows, to toggle between them, and even to be able to jump straight to a particular window. Byobu has all these options ready for you. Use 'Ctrl-a n' and 'Ctrl-a p' to switch to the next or previous window in the list, by number, and 'Ctrl-a #', where # is a number from 0 to 9, to switch to the corresponding window. To get a full-screen list of windows, use the combination, 'Ctrl-a " '; navigate this list with the arrow keys and pick a window to activate by pressing Enter when it is highlighted (**Fig 3**).

12 Name or rename a window

You can name or rename a window with relative ease, even as you are using the window. Hit the key combination of 'Ctrl-a A'. You will see a prompt where you can see the old name of the window. Remove it and set your new name there. Hit Save for it to get saved.



■ Rename a window to make it easier to keep track of it

13 Default windows

In a previous step we looked at the process of creating a new window from the

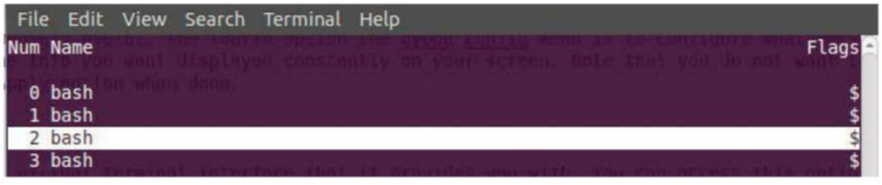
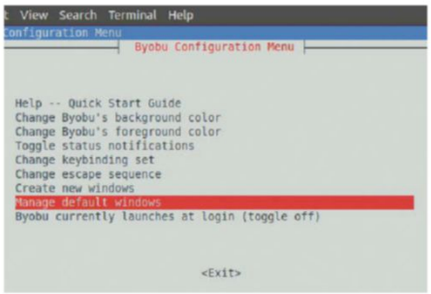


Fig 3 Moving between windows A list of open windows

Byobu menu. There was an option there to add the window to the default list of windows. This is a list of windows that will get launched by default. You can view and manage this list by going to the 'Manage default windows' option in the menu. Here you can select the windows you want launched when you launch the Terminal.



Manage the list of windows you want launched by default

14 Manual configuration

If you would like a bit more control over the configuration of Byobu, you can access the configuration files from the hidden folder called .byobu under your home directory. Here you will find a number of configuration files such as the status file where you can easily configure your preferred settings manually (**Fig 4**).

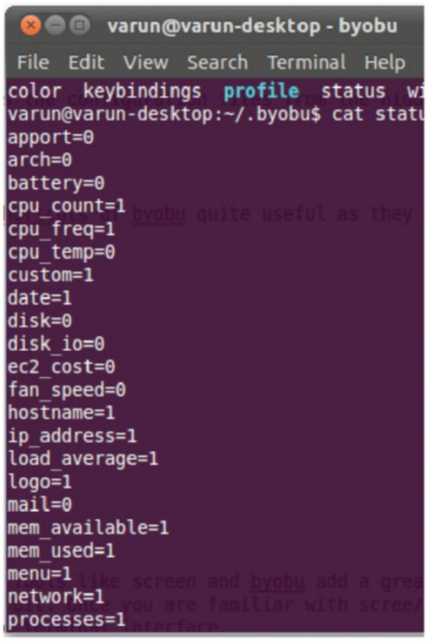


Fig 4 Manual configuration The manual configuration is a good option for advanced users, offering extra control

15 Moving around

If you are familiar with the Vim or Vi text editors, you will find the navigation shortcuts of Byobu quite useful as they are pretty much the same. **Here is a list of a few of the most useful movement keyboard shortcuts...**

h Move the cursor left by one character
j Move the cursor down by one line
k Move the cursor up by one line
l Move the cursor right by one character
0 Move to the beginning of the current line
$ Move to the end of the current line
G Moves to the specified line (defaults to the end of the buffer)
/ Search forward
? Search backward
n Move to the next match, either forward or backward

16 Keyboard shortcuts

Here are some useful keyboard shortcuts that you should familiarise yourself with. Tools like Screen and Byobu add a great deal of efficiency to your workflow. However, if you are dependent on using the mouse for menus, things might slow down a bit. **Once you are familiar with Screen/Byobu keyboard shortcuts, you will see a noticeable change in the speed and efficiency with which you go about working on the Terminal interface.**

F2 Create a new window
F3 Move to previous window
F4 Move to next window
F5 Reload profile
F6 Detach from this session
F7 Enter copy/scrollback mode
F8 Re-title a window
F9 Configuration menu
F12 Lock this Terminal

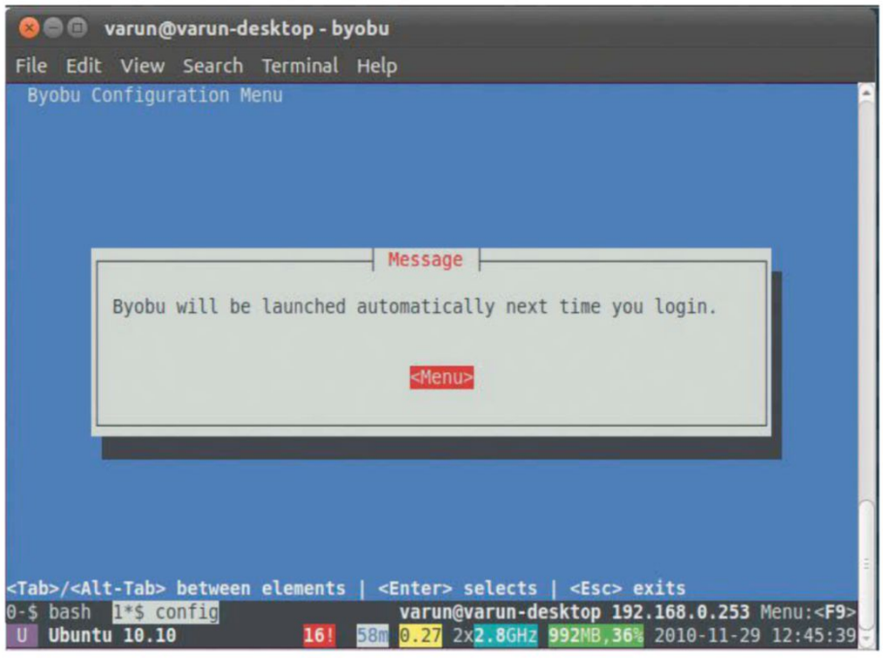
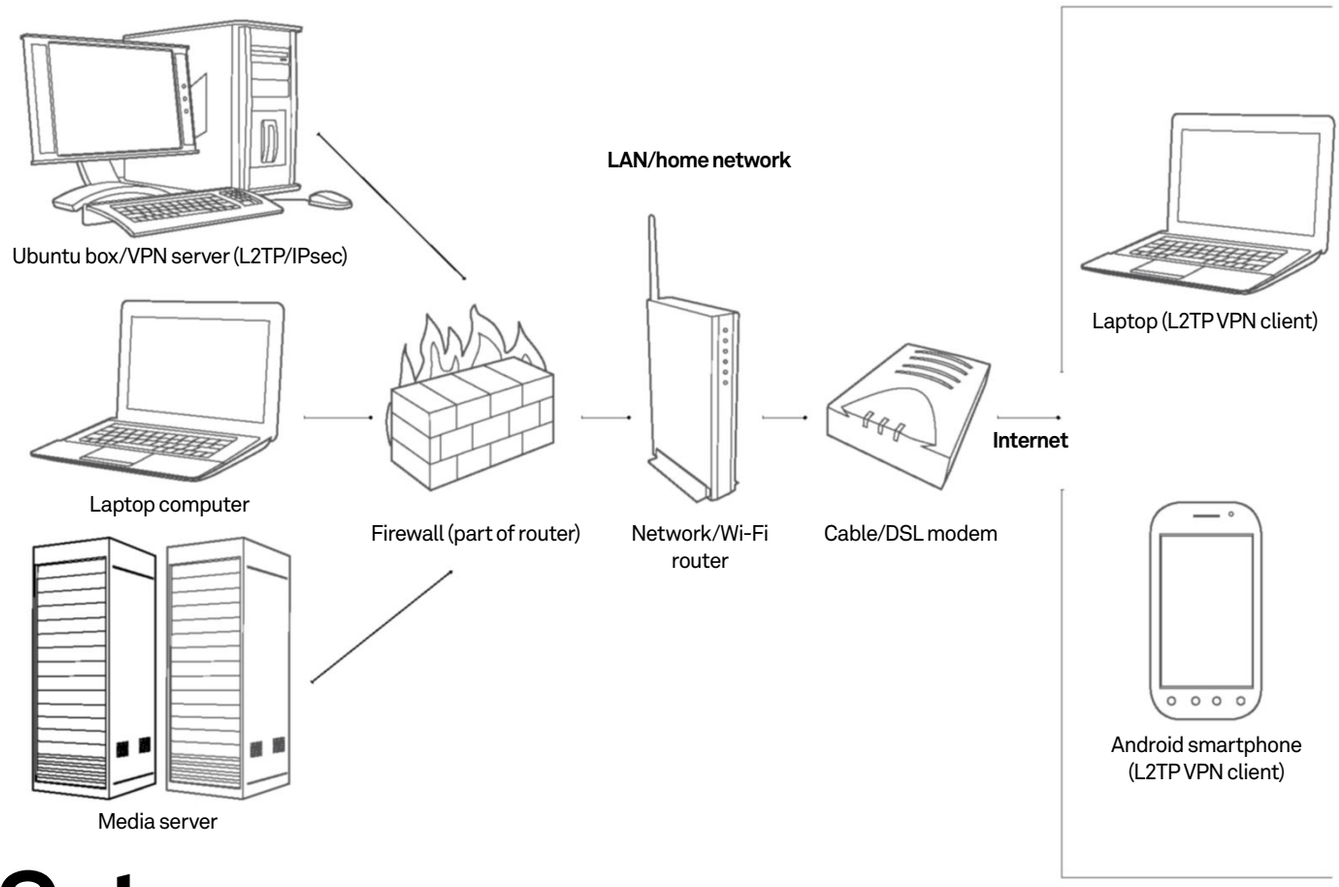


Fig 5 Launch Byobu by default Set Byobu as your default Terminal interface

17 Launch Byobu by default

If you want Byobu to be activated automatically every time you launch the Terminal window, you will need to make a change to the settings. Hit F9 after entering the Byobu interface. You will be presented with a menu. Choose the last option in the menu. Hit Enter to toggle the option whether to log into Byobu upon login or not. Then in the gnome-terminal application, go to Edit> 'Profile preferences' and select the 'Title and Command' tab. Check the 'Run command as login shell' option (**Fig 5**).

In this article, we looked at a few of the options that the wonderful tool Byobu gives us. However, it is very difficult to cover all the useful options. Therefore we strongly urge you to spend some time reading the application's man page. It will greatly help you enhance your workflow.



Set up your own personal VPN server

Learn how to access your personal network and data securely from anywhere on the internet

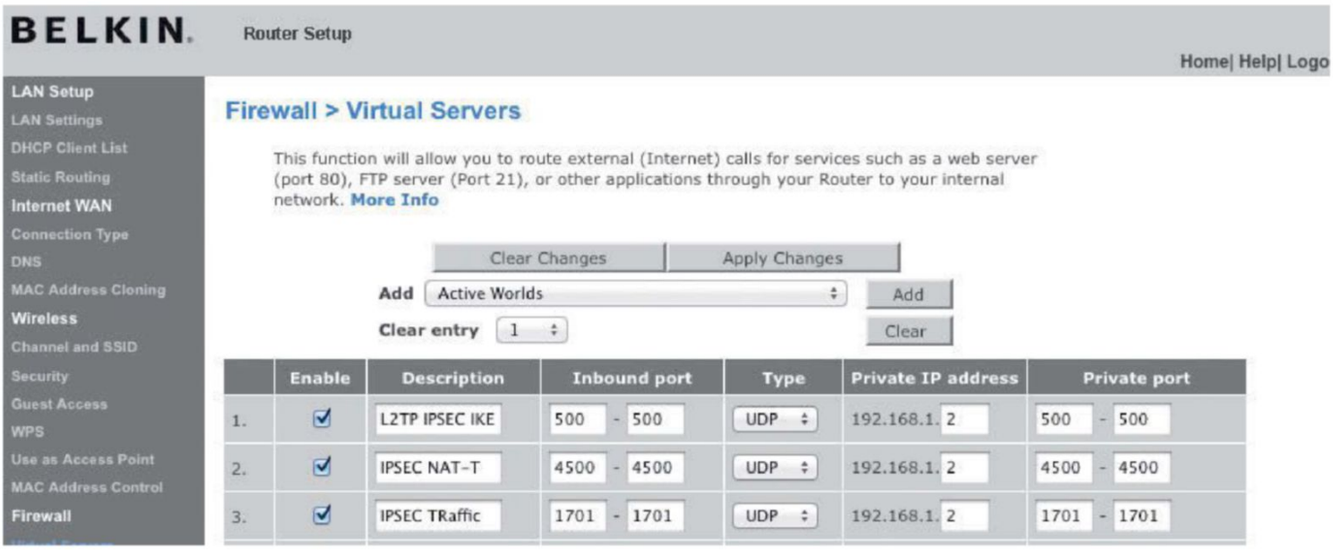
Resources

A recent Linux distro
Support packages (links in guide)

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

There are two types of network that we use in everyday life. One is the internet (or as they say in the textbooks, WAN – wide area network) and the other is your home (or office) network, also referred to as a LAN (local area network) or intranet. Both of these networks coexist simultaneously. A very common example would be your home network which includes a cable (or DSL) modem, a Wi-Fi router, your home PCs, media servers, smartphones etc. All these devices connect to the internet via a gateway device, often a network router. There are a lot of reasons behind why our home networks are designed this way, but the most important reason is to enable the sharing of one internet connection by multiple devices. Other reasons include the security and management aspects. However, there are also downsides to this approach. If you are outside of your home network, you will not be able to make use of any resources available on it. For example, you won't be able to access your music collection if you are not on your home network. That is where a VPN, or virtual private network, comes in. As the name suggests, it allows you to create a virtual private network on the public internet, so you can access your home network while way from home.



Planning

In this tutorial we will be building a VPN server using Ubuntu that will allow us to access our home network from the internet. Before we proceed any further, let's plan a sample scenario. This will help you to adapt the setup to your own home network.

For our sample network (which we believe is how most of you will have your home network configured), we have the following:

• LAN network configuration: Your network uses a 192.168.1.XX series of IP addresses with subnet mask 255.255.255.0 . With the exception of the Ubuntu box, all the devices in your network are auto-configured via the DHCP server (available on the network router).

• An Ubuntu box: This is your typical desktop/laptop which we will be configuring to act as the VPN server. This box will have a local static IP address, 192.168.1.2.

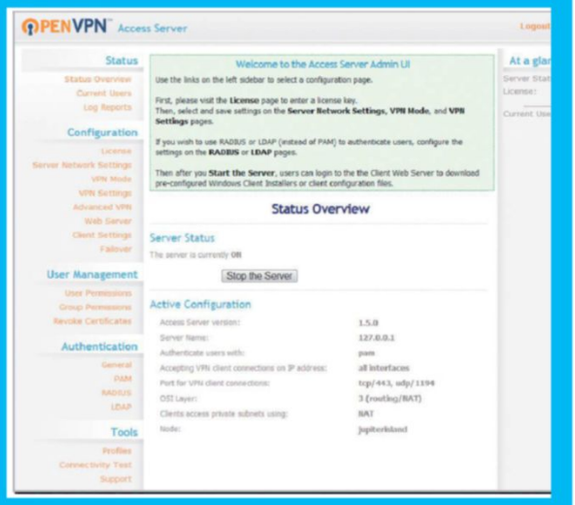
• Network router: The router acts as a gateway device, with the IP address 192.168.1.1. This device is connected to your cable/DSL modem (not available if you have an integrated device). We will not be interacting with the modem directly, as most of it should be configured by the ISP. However, we need the external IP address handy. This is the IP address which is accessible from the internet. Most ISPs provide two types of IP addresses: static or dynamic. Having a static IP address is very helpful in

"In this tutorial we will be building a VPN server using Ubuntu that will allow us to access our home network from the internet"

OpenVPN

We decided not to cover OpenVPN because of the lack of support on the Android and iOS platforms, but that does not make it any less of a VPN platform. OpenVPN is a fully featured SSL VPN that implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol; supports flexible client authentication methods based on certificates, smartcards, and/or username/password credentials; and allows user- or group-specific access control policies using firewall rules applied to the VPN virtual interface. So you see, in a way OpenVPN is more sophisticated and has more features than the L2TP/IPsec-based VPN setup.

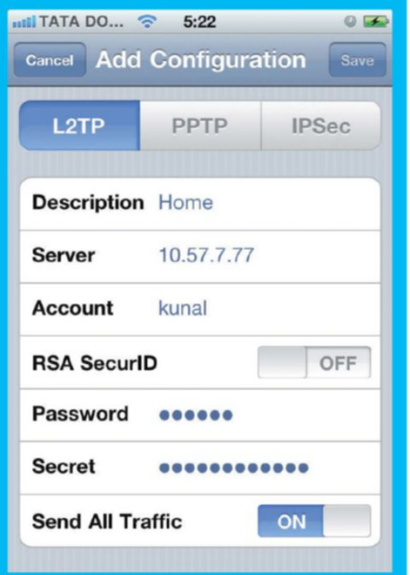
Like L2TP/IPsec, OpenVPN is supported (Community Version) on Ubuntu boxes. The OpenVPN project also provides a commercial web GUI (available on **OpenVPN.net**) which might be easy to set up and maintain for novice users.



Configuring L2TP/IPsec client on iOS devices

Perform the following steps to configure the L2TP/IPsec client on an iOS device:

1. Go to Settings>General>Network>VPN> Add VPN Configuration
2. Select the VPN type as L2TP.
3. Fill in the Description to identify the VPN connection.
4. Set the Server as the external IP address of your network, as assigned by your ISP.
5. Set any username in the Account field, as our configuration for xl2tpd does not check for it.
6. Leave RSA Secure ID as Off.
7. Enter the Password as entered in the file /etc/xl2tpd/l2tp-secrets.
8. Enter the Secret as the PSK set in the /etc/ipsec.secrets file.
9. Save it. You can now start this VPN connection by going to Settings>VPN and toggling the VPN switch to On. Make sure that the correct VPN connection is selected while starting the VPN connection.



L2TP/IPsec configuration on an iOS device

this case because it is easy to configure and remember. Dynamic IP addresses should also work, but you should be able to find out your external IP address once you are connected and make sure whether it is reachable from the outside internet.

• VPN client devices: These are devices which have the designated VPN client. In this case we have chosen to go with L2TP/IPsec as it is widely supported by many platforms, including Linux, Mac, Android, iOS and Windows. We won't be going in detail about configuring the client, but it is very straightforward. Most of the time it is only a matter of punching in the right IP address and credentials.

1. Setting up the virtual servers (or IP port forwarding)

Since all the outgoing connections are directly handled by the router itself and not the Ubuntu box, the router should be able to forward the necessary client connection request to the Ubuntu box. This can be set up by configuring IP port forwarding. On our test router (Belkin N+) this setting is called Virtual Server. Open your router's IP port forwarding page and **configure it as follows :**

```
Port UDP 500 to 192.168.1.2 UDP 500
Port UDP 4500 to 192.168.1.2 UDP 4500
Port UDP 1701 to 192.168.1.2 UDP 1701
Port TCP 50 to 192.168.1.2 TCP 50
```

2. Disable firewall on Ubuntu box

As we are using our network router as a firewall, we do not want to create any conflicting settings on the Ubuntu box. So make sure that you have disabled the firewall on it. If required, create more rules on the router.

To manage firewalls in Ubuntu the easy way, you can install a tool called Gufw.

```
$ sudo apt-get install gufw
```

Open Gufw and uncheck the Enabled button to disable the firewall (**Fig 1**).

> **L2TP needs to be paired with an IPsec implementation to provide encryption**

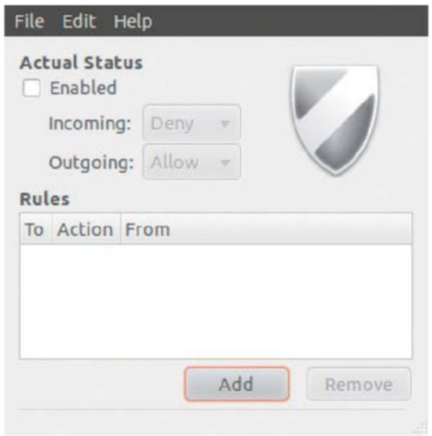


Fig 1 Gufw is an easy-to-use Ubuntu firewall

3. Installing VPN support packages

Our VPN setup requires the following three packages on the server side...

xl2tpd: This is a Layer 2 Tunnelling Protocol (L2TP) daemon. You can install it using the command 'sudo apt-get install xl2tpd'.

Strongswan: L2TP on its own does not provide any encryption. It needs to be paired with an IPsec implementation to provide encryption. Strongswan is an implementation of IPsec for Linux. You can install Strongswan by using the command 'sudo apt-get install strongswan'.

PPP: This is a basic point-to-point protocol implementation. PPP should already be installed on your system; if not, you can install it using the command 'sudo apt-get install ppp'.

4. Configuring IPsec (Strongswan)

IPsec is a communication protocol based on IP. IPsec appends security of communication to IP. TCP/IP and UDP/IP gain security from IPsec without being aware of it. IPsec provides encryption, integrity insurance and authentication of data. In this case we will be using these features in conjunction with L2TP.

The primary configuration file for IPsec is /etc/ipsec.conf. Edit it to look like the following:

```
@config file:/etc/ipsec.conf
 config setup
       virtual_private=%v4:10.0.0.0/
8,%v4:192.168.0.0/16,%v4:172.16.0.0/12
       nat_traversal=yes
       protostack=netkey
       oe=no
  # Replace eth0 with your network
interface
       plutoopts="--interface=eth0"
  conn L2TP-PSK
```

```
        authby=secret
        pfs=no
        auto=add
        keyingtries=3
        dpddelay=30
        dpdtimeout=120
        dpdaction=clear
        rekey=yes
        ikelifetime=8h
        keylife=1h
        type=transport
  # Replace IP address with your local
IP of the Ubuntu Box
        left=192.168.1.101
        leftnexthop=%defaultroute
         leftprotoport=17/1701
  # Replace IP address with your VPN
server's IP
        right=68.68.32.79
        rightprotoport=17/1701
```

Pay attention to the comments and update the values to reflect your network environment.

5. Configuring the IPsec pre-shared key (PSK)

The IPsec pre-shared key is used to authenticate the IPsec-based packets. It is stored in the following format. **You will need to create this file in the /etc directory.**

```
@/etc/ipsec.secrets
your.ip.goes.here %any:      PSK
"yoursharedkeygoeshere"
```

Here, '%any' allows any host to connect with the given pre-shared key. You can also define a different PSK for each host; for each, insert the expected host address in the place of '%any'.

6. Configuring the L2TP daemon

Now is the time to configure the L2TP daemon. Much of the configuration is stored in the /etc/xl2tpd/xl2tpd.conf file. **Edit it as follows:**

```
@config file:/etc/xl2tpd/xl2tpd.conf
/etc/xl2tpd/xl2tpd.conf:
[global]
debug network = yes
debug tunnel = yes
[lns default]
ip range = 192.168.1.54-192.168.1.87
local ip = 192.168.1.53
require chap = yes
refuse pap = yes
require authentication = yes
name = mydomain.com
ppp debug = yes
pppoptfile = /etc/ppp/options.xl2tpd
length bit = yes
@
```

7. Configuring the L2TP password

In this step we will be configuring the L2TP password. Please keep in mind that this password is different from the pre-shared key mentioned in the IPsec step. **To configure the L2TP password, open /etc/xl2tpd/l2tp-secrets and edit it as follows:**

```
@/etc/xl2tpd/l2tp-secrets
*       *       <l2tppassword> *
```

This password will act as the VPN user password for any VPN user.

8. Configuring the L2TP options

In this step we will configure the general L2TP options. General options are stored in /etc/ppp/options.xl2tpd. **Edit this file to reflect the following :**

```
ipcp-accept-local
ipcp-accept-remote
ms-dns <your DNS IP Address, For Google
DNS use 8.8.8.8>
noccp
auth
crtscts
idle 1800
mtu 1410
mru 1410
nodefaultroute
debug
lock
proxyarp
connect-delay 5000
```

9. Applying the configuration

Now that we have configured everything, it is time to apply the changes. This can be done by restarting the services associated with IPsec and the PPTP (L2TP).

To restart the IPsec service :

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

To restart the L2TP service :

```
$ sudo /etc/init.d/xl2tpd restart
```

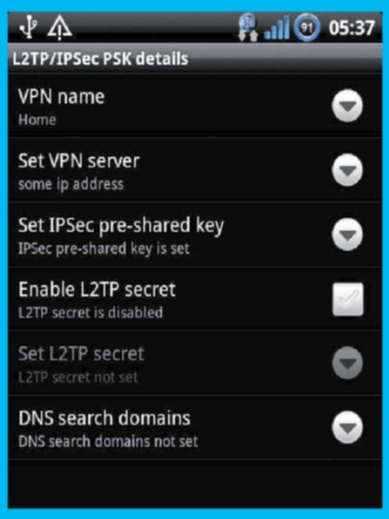
So you see how easy it is to build a VPN server using your favourite Ubuntu box. Configuring the VPN server does require some networking skills on your part, but it is not difficult. If it is does not work for you, try again or ask on the **LUD** forum (**www.linuxuser.co.uk/forum/**) for any help with your particular network configuration. We hope you enjoy accessing your home network while away from home.

> **"The PSK is used to authenticate the IPsec-based packets"**

Configuring the L2TP/IPsec client on an Android device

Perform the following steps in order to configure the L2TP/IPsec client on an Android device:

1. Go to Settings>Wireless & Network> VPN Settings>Add VPN>Add L2TP/IPsec PSK VPN.
2. Fill in the VPN Name to identify the VPN connection.
3. Set VPN Server as the external IP address of your network, as assigned by your ISP.
4. Set IPsec pre-shared key as the PSK set in the /etc/ipsec.secrets file.
5. Tap Menu and then tap Save to create the connection. You can now start this VPN connection by going to Settings> VPN and then tapping the created VPN from the VPNs list. You will be asked for a username and password. Enter any username and password, as entered in the file /etc/xl2tpd/l2tp-secrets.



Build your own multimedia NAS

Network-attached storage (NAS) devices are incredibly versatile and can serve both as a file server and multimedia server. In this guide you'll learn how to build the perfect open source multimedia NAS box

Every household now has its own network. These networks are usually connected to an ever increasing selection of devices such as PCs, laptops, smartphones, TVs, game consoles, Blu-ray players and other assorted media players. These devices often need to share data, but most of the time we don't want to share data physically – with files and folders spread all over the network – but from an easily managed, central location. That is where network-attached storage (NAS) comes in. An NAS is a standalone storage appliance that provides file-level access via any standard network connection. An NAS is typically a very low-powered appliance when compared to a traditional server, and as such costs much less to run on a 24/7 basis.

In this tutorial we will be building our own NAS box designed to provide useful features and important characteristics...

Green: Since an NAS box will be running 24/7, we will make sure that it consumes less energy.

Wide protocol support: We will have support for CIFS (via Samba), TFTP, FTP, NFS etc.

DLNA support: DLNA and UPnP support will enable our NAS box to stream media to a wide variety of multimedia devices, such as TV, set-top box, Blu-ray player etc.

Wide file system support: Ext2/ext3 support to ensure our Linux formatted disk support. We would also like to have support for Windows-based file systems such as NTFS and FAT32.

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Installing FreeNAS

1. Preparing

Burn the ISO file to a CD/DVD. Format the USB drive to FAT32. Connect the USB drive to the system and boot from the CD/DVD. Upon booting from CD/DVD, you will have the option to continue with the default boot or boot with special options.

CD/DVD Boot Prompt

Special options are helpful when you are facing issues with the default mode.

Continue with the default boot.

2. Starting installation

After a few moments you will get the FreeNAS live screen (**Fig 1**). It allows you to do a range of things such as configuring interfaces, setting up LAN IP address etc. At this point we are interested in option 9) 'Install/Upgrade to hard drive/flash device, etc'.

3. Installing

Now we will do the final installation. From the next screen select 'Install 'embedded' OS on HDD/Flash/USB'. This option will install the embedded version of the FreeNAS on your system. If you want a more fully fledged installation, you can select option 2 or 3. Next, you will be presented with a warning screen which will advise you about what's going to happen. On the next screen you will be prompted with a list of optical drives detected

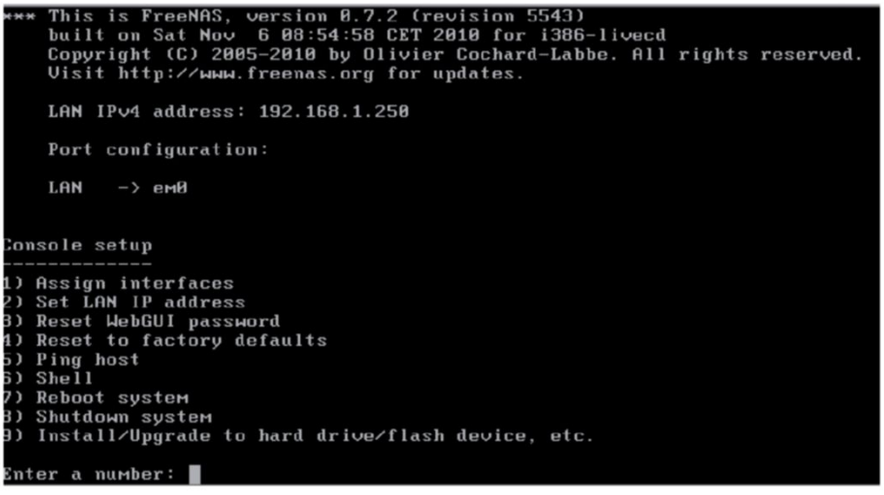


Fig 1 Starting installation FreeNAS live screen

Resources

Keeping our features in mind, let's move on to our software and hardware needs. We will try to keep things minimal here. Keep in mind that you do not need the latest and greatest hardware to run a NAS server. If you have an old PC eating dust, this provides the perfect opportunity to dust it down. **If you don't have one, not a problem – just make sure you have met the following as minimum requirements…**

FreeNAS FreeNAS is an embedded open source NAS (network-attached storage) distribution based on FreeBSD. This software is available for free. We wish we had a Linux-based distribution, but as of now there is no good Linux-based NAS distribution available. OpenMediaVault is a Linux-based NAS distribution created by the original FreeNAS developers, but it's yet to be released. FreeNAS can be obtained from http://freenas.org/

Motherboard & CPU Ideally you should go for a mini or smaller form factor motherboard. You can also go for a motherboard-CPU combination to save money. Our recommendation is the Intel Desktop Board D510MO with integrated Intel Atom processor (£57.60, from Amazon). The D510M provides full compatibility with FreeNAS and it has Gigabit Ethernet support (crucial for a box of this nature). It has two SATA ports and seven USB ports. This board consumes a maximum of 52.5W in its standard configuration

HDD This is purely driven by your own requirements. However, as a general rule of thumb, the more the better. If you want data redundancy, you may want to add more drives. You will also need to buy a separate SATA controller (or RAID controller) card if you want to connect more than one internal drive

Flash drive This will contain the installation of FreeNAS. On a separate note, you should also make sure that you are able to boot from the Flash Drive (or a USB stick).

RAM One 2GB RAM stick will be sufficient. To keep your system green, buy RAM sticks rated around 1.35V

Misc You will also need other things such as a CD drive (required only for installation), power supply, mouse, keyboard and case etc. If you want to have Wi-Fi support, you will also need to buy a Wi-Fi USB adaptor

This tutorial is built with the following assumptions:

1. The FreeNAS distribution will be installed on a dedicated USB disk and not on the internal hard drive.
2. This installation does not deal with the RAID or other redundancy related configurations.
3. We are using FreeNAS version 0.7.2 (filename FreeNAS-i386-LiveCD-0.7.2.5543). This file can be obtained from **http://sourceforge.net/projects/freenas/files/stable/0.7.2/FreeNAS-i386-LiveCD-0.7.2.5543.iso/download**.
4. FreeNAS 0.7.2 is based on FreeBSD 7.2 . Any hardware supported by FreeBSD 7.2 will work with FreeNAS without any issue. You can look at the hardware compatibility list at **www.freebsd.org/releases/7.2R/hardware.html**. For hardware which is not supported by the FreeBSD 7.2 directly, you can use the unstable version or compile a build from the SVN repository with the necessary patches.
5. You have an old PC to work with or you know how to assemble a system on your own.

Fig 2 Installing FreeNAS Installation

> **An NAS is a standalone storage appliance that provides file-level access via any standard network connection**

on your system. Select the one which contains the FreeNAS distribution disc. Finally, select the location where you want to install the FreeNAS distribution. Select the USB flash drive in this step. The installation will now begin (**Fig 2**).

After completing the installation, remove the installation disc and reboot the system. Make sure that your system boots from the USB drive after the restart.

Configuring FreeNAS

4. Network configuration

Upon system startup, select option 2 for setting up the LAN IP address. The next screen will ask if you want to use a DHCP server to configure the interface. As the NAS box is a 24/7 appliance, we would like to have it a permanent IP address. Select 'No' to continue with the network configuration. Enter the required details to set up a manual IP address for the system (**Fig 3**).

5. Using the WebGUI interface

WebGUI is an HTTP service of FreeNAS which enables the configuration of almost every aspect of FreeNAS via an easy-to-use and intuitive user interface (**Fig 4**). Like any common website, WebGUI runs off port 80 and be accessed by navigating to the URL http://<FreeNAS IP Address>. The default username is 'freenas' and the password is 'admin'.

6. Configuring the storage

Now it is time to add storage to the NAS server. Open WebGUI, go to Disks>Management and Click '+' to add a new disk. The 'Add' page will open. Here you will be provided with the list of disks available on the system. Enter the description and click Add (**Fig 5**). You will need to repeat this step for each hard disk that you have added to the system. Go back to the Disks Management to view the online status of the attached disks (**Fig 6**).

Fig 3 Network configuration FreeNAS LAN IP address configuration

> **Note**
> **Do not add the disk on which you have installed FreeNAS, otherwise you may create an unusable system.**

Fig 4 Using the WebGUI interface FreeNAS WebGUI interface

> **"WebGUI is an HTTP service of FreeNAS which enables the configuration of almost every aspect of FreeNAS via an easy-to-use and intuitive user interface"**

7. Formatting the disks

In this step we will format the disks attached to the FreeNAS System. Go to Disks>Format. Select the disk you want to format from the Disk drop-down list. Select file system as UFS, enter a volume label to identify the disk and click 'Format disk' to start formatting the disk (**Fig 7**).

Depending upon the disk size, formatting can take a considerable amount of time.

8. Managing services

FreeNAS supports a wide range of services that make it an incredible solution for a NAS box. In this step we will configure a few of the important services .

CIFS/SMB: CIFS (Common Internet File System), or SMB, is a network protocol particularly popular in the Windows world. It is used by most standalone media players as well as in home theatre software such as Boxee. To enable this service, got to Services>CIFS/SMB. This will open the CIFS/SMB Settings page (**Fig 8**). Check Enable to enable the service. Change NetBIOS name and Workgroup according to your requirements. You can keep Authentication as Anonymous or Local User or Active Directory. For maximum security it is recommended that you use Local User authentication. Click on Save and Restart. Click the Shares Tab, then the '+' icon to add the shared path.

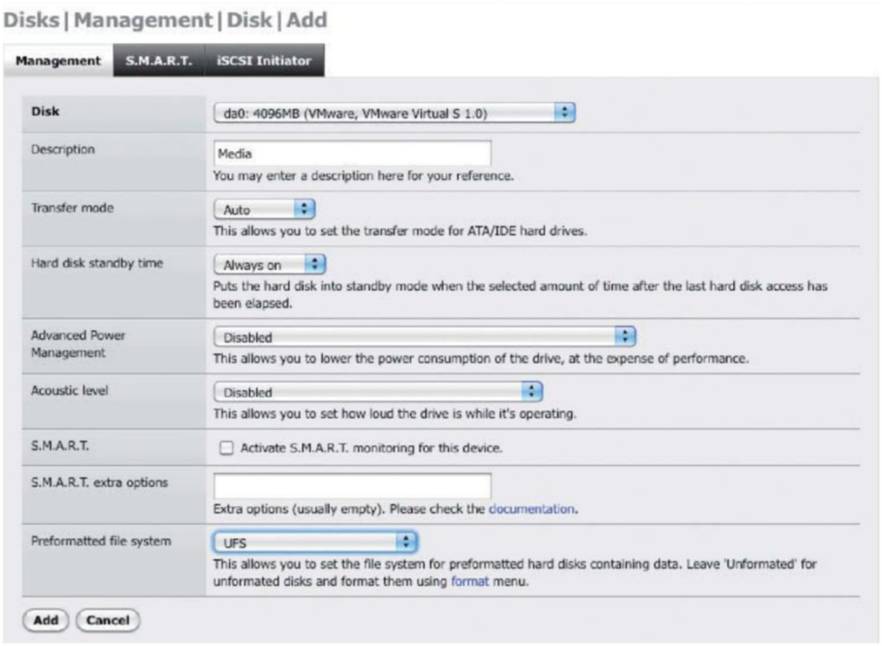


Fig 5 Configuring the storage Adding a disk to the FreeNAS system

Fig 6 Configuring the storage Disk status

SSH: SSH (Secure Shell) enables remote administration of the FreeNAS server. It also enables file transfer over SFTP. To enable the SSH service, click on Services, then SSH. This will open the SSH Settings page. Check Enable to enable the SSH service.

UPnP: FreeNAS features a built-in UPnP AV media server (or DLNA server). You can use the UPnP AV media server to store and stream digital media content to a UPnP AV client (or DLNA client) device. These devices include modern televisions, set-top boxes, smartphones etc. To enable the UPnP service, click Services>UPnP. This will open the UPnP Settings page. Check Enable to enable the UPnP service. **Now, enter the following information…**

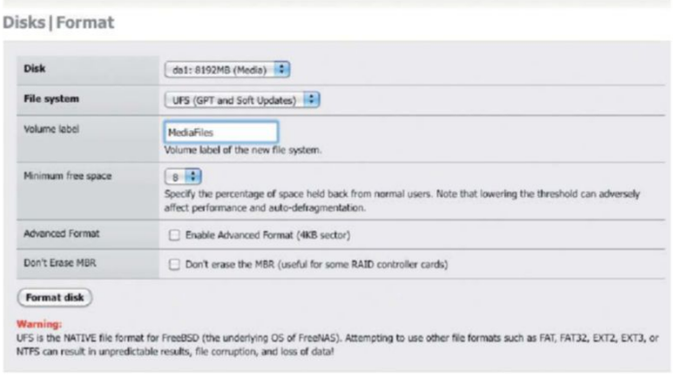


Fig 7 Formatting the disks Disk formatting page

Fig 8 Managing services The CIFS/SMB configuration page

Database directory: Specify where you want to store the catalogue of the media files.

Content: Add all the media directories. This will make the FreeNAS service scan and catalogue the media file information.

Profile: FreeNAS supports a variety of profiles. You should use the one which matches closely to the client you are using. FreeNAS profile support includes DLNA, Sony PlayStation 3, Microsoft Xbox 360 and many others.

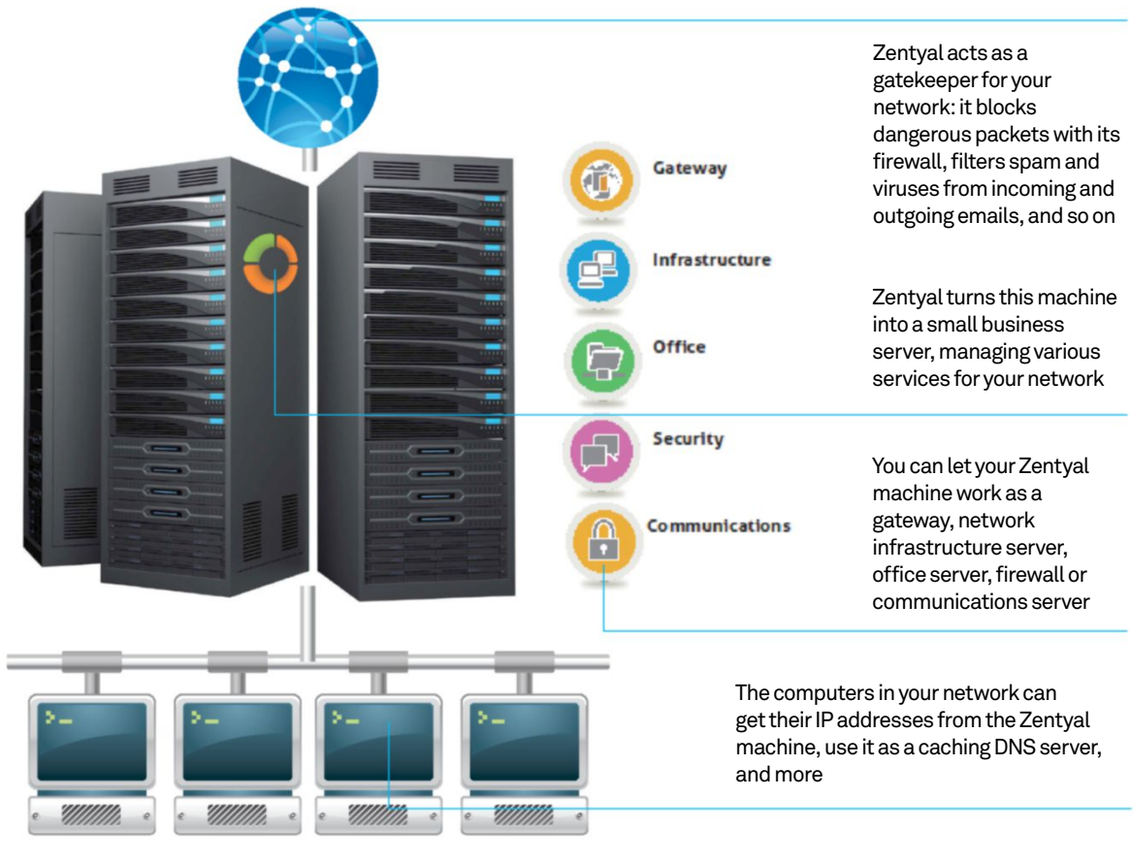
Transcoding: Here you get a chance to enable transcoding. Transcoding allows the dynamic encoding of the media to provide the maximum compatibility with the clients. For example, you may want to enable transcoding if you want to stream MKV files to an Xbox 360 or PS3. However, the same is useless for the D-Link Boxee Box as it has built-in support for MKV files.

WebGUI: This option enables the support for FUPPES WebGUI for the UPnP service.

That's it, you are done. You have built your multimedia NAS box. You can also do lot of crazy things using your NAS box, such as running a BitTorrent client or creating an iTunes streaming server. There is no limitation on how much you can do with it. So if you are in the market for an NAS box, why not build one yourself?

Build a Linux server for your small business

Do you need a Linux server to manage your small business network? Koen Vervloesem thinks you should try Zentyal



Advisor

Koen Vervloesem has been writing about free and open source software, and IT in general, since 2000. He has master's degrees in computer science and philosophy and can be reached at koen@vervloesem.eu

Zentyal began in 2004 under the name 'eBox Platform'. Its focus has always been on developing an intuitive web interface to unlock the possibilities of a Linux server for small businesses. Zentyal 2.0.3 is based on Ubuntu Server 10.04 LTS and it can be used as a gateway, a firewall, a communications server, and so on. It offers all the basis services a small business needs: DNS, DHCP, NTP, a CA (certificate authority for SSL), a web server, an FTP server, a firewall and router, quality of service (QoS), network authentication with RADIUS, an HTTP proxy, virtual private networks (VPN), intrusion detection, a mail server (including webmail), file and printer sharing, groupware, a directory service with LDAP, instant messaging using Jabber, VoIP and more. All of this is free-to-use, but Zentyal development is funded by the company Zentyal, which also offers extra management tools and services. So when you decide to implement Zentyal in your company and you want some extra training or maintenance support, you can buy this from Zentyal or a local Zentyal partner.

Resources

Zentyal http://www.zentyal.org

Zentyal documentation http://doc.zentyal.org/en/

1. Hardware requirements

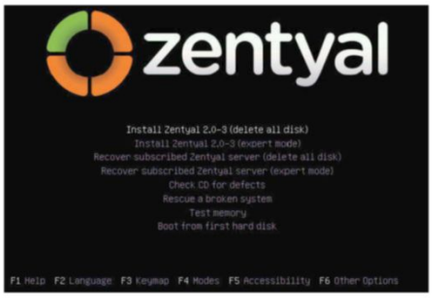
First check if Zentyal runs on your hardware. This means that Ubuntu Lucid 10.04 LTS (kernel 2.6.32) should support the hardware. You can check the Ubuntu Linux Hardware Compatibility List for supported components or the list of servers certified for Ubuntu 10.04 LTS, both on Ubuntu's website. Of course the CPU, RAM and disk space you need depend on the modules you install and the number of users.

2. Install on Ubuntu

You can install Zentyal on top of an existing Ubuntu Server Edition installation. If you want to do this, you have to add the official Zentyal repository with the command 'sudo add-apt-repository ppa:zentyal/2.0', and after a 'sudo apt-get update' you can install Zentyal with 'sudo apt-get install zentyal'.

3. Install standalone

However, the preferred way is to install Zentyal from scratch with the ISO, which installs a preconfigured Ubuntu Server 10.04 with Zentyal on top of it. We won't go into detail about the installer, as the installation process is identical to that of Ubuntu Server.



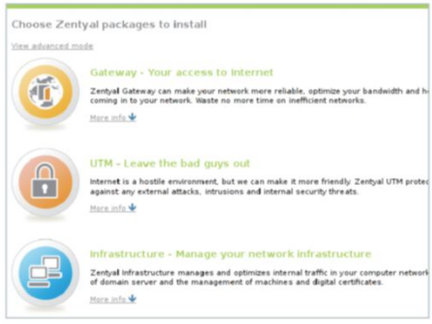
4. Graphical interface

Although Zentyal is a server platform, it installs a lightweight desktop environment that functions as an administrative interface (**Fig 1**). The first time you boot your Zentyal installation, the desktop shows itself and opens the web interface, which is also accessible from other machines on your internal network. If possible, you should do the initial configuration locally on the Zentyal machine, because otherwise you lose access during the network configuration.

Fig 1 Graphical interface

5. Initial configuration

When you open the web interface for the first time (log in with the username and password you chose during the installation process), a configuration wizard shows various installation profiles. You can select one or more of these profiles (they get a green ring when selected) to install a group of packages that implement specific functionality. Alternatively, you can pick individual Zentyal modules using the advanced mode.

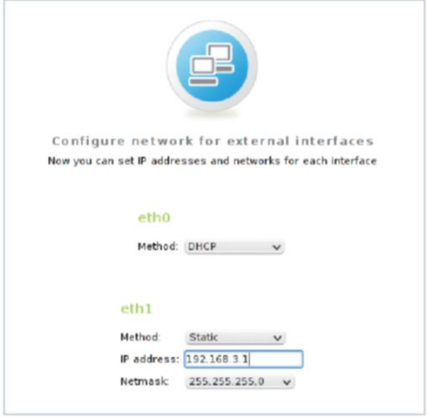


6. Gateway

We're choosing to install the Gateway profile, so our Zentyal machine can function as our network's router, giving access to internet and implementing a couple of convenient network services. So we click on the Gateway icon to select it and then on the Install button. You can see a list of the selected packages and you will be asked whether you want to install recommended components. Click on OK to install the packages.

7. Configure network

Now you have to configure the network interfaces of your computer. For each network interface, specify whether it is connected to an external network such as internet or to an internal network (the local network that your Zentyal box will be a router for). Zentyal will use this information for its firewall policies. Click Next and then choose how to configure the IP address for each interface. Save your changes at the end.



8. Dashboard

Now access Zentyal's dashboard on its web interface (**Fig 2**, overleaf). This shows you the status of your machine (such as the system load, uptime and the number of users), its network traffic and a list of the running services. This is also the place where you can quickly restart any service. If you click on 'Configure widgets', you can tailor the information that is shown to your taste by dragging and dropping widgets.

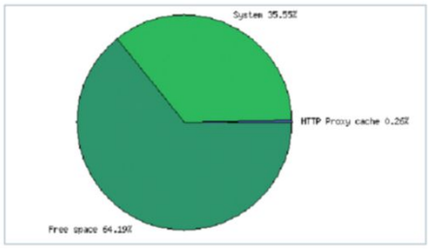
9. Enable modules

The page 'Module Status' shows a list of all available modules. Tick a module's checkbox

if you want to enable it. The first time that you enable a module, Zentyal asks you to accept a set of actions to carry out, such as overwriting configuration files. After you have accepted this, you have to save the changes by clicking on 'Save changes' in the top bar. If a module depends on another one in order to work, you can't enable it until the dependencies are enabled.

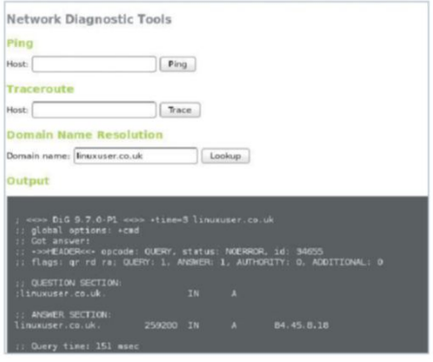
10. System

The System menu gives you access to some essential system preferences. This is the place where you can change your user's password, the hostname, the port number where the web interface is accessible, the date/time and timezone, and more. You can also halt or reboot the machine here, look at some graphs about the disk usage, make a backup of the current configuration or restore the configuration from a backup file.



11. Network

The Network menu lets you configure all things related to the network. For instance, you can change the configuration of your network interfaces here, you can add domain name servers and configure a dynamic DNS service, you can set up internet gateways, static routes and an HTTP proxy, and you can even configure WAN failover and balance traffic (if you have two gateways configured). Moreover, the 'Diagnostic Tools' page lets you use ping, traceroute and hostname lookups to diagnose network issues.



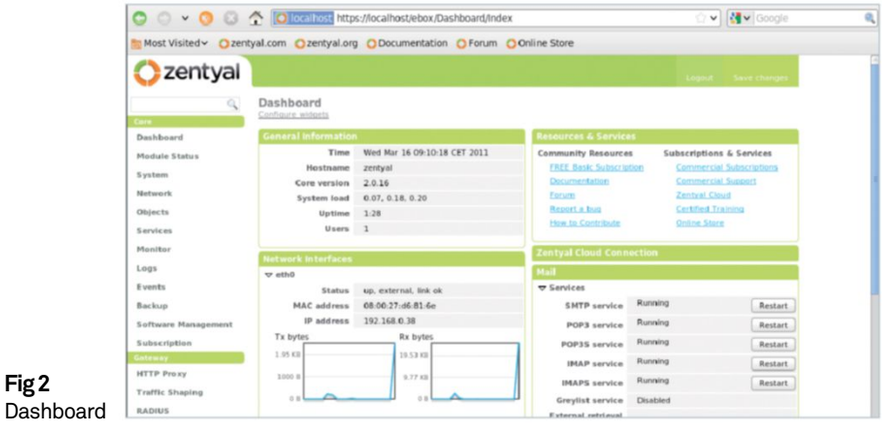
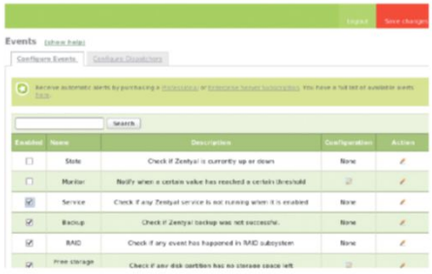


Fig 2
Dashboard

12. Events

On the Events page, you can configure how Zentyal informs you about events that happen. By default, it only informs you if a backup was not successful, but you can enable other events at will. For example, it's wise to be informed as soon as possible if any event has happened in the RAID subsystem or if a disk partition has no storage space left. In the latter case, you can choose the threshold that triggers the event, which is by default 10% of the disk space. You can also let Zentyal check if any service is not running when it is enabled.



13. Dispatchers

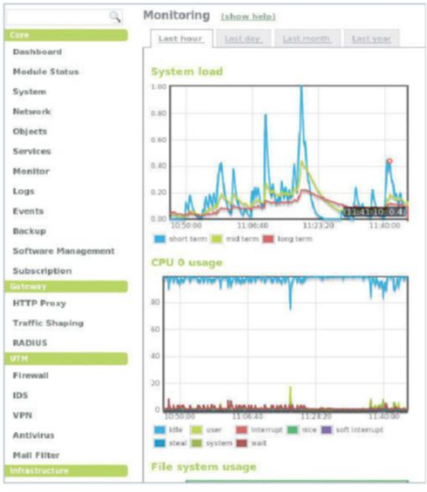
If you click on 'Configure Dispatchers' on the Events page, you can choose the means by which you want to be informed. By default, Zentyal warns you in the most unobtrusive way – log files – but you can also enable RSS alerts, emails to the administrator, or Jabber messages. Each of these dispatchers can be configured in more detail, eg the subject and email address for the mail dispatcher.

14. Logs

If something ever goes wrong, the first place where you should look is the log files. Zentyal shows a nice query interface for the log files in the Logs menu, where you can consult a full or a summarised report for various services, such as the firewall, DHCP server, mail, VPN, HTTP proxy, and so on. You can also tell Zentyal to purge logs older than a predefined period (by default logs are saved for one week).

15. Monitor Zentyal's health

The Monitor page shows you a couple of graphs, eg with the system load, the CPU usage, the file system usage and the physical memory usage. These graphs can be used to identify problems or check if your hardware is underpowered. For instance, if the system load, CPU usage or memory usage is constantly too high, you definitely need a hardware upgrade. By default, the Monitor page shows you a graph of the last hour, but you can easily switch to a daily, monthly or even yearly graph; these are especially interesting for trending. For example, in the daily graph you can see at which times the peak moments are, and the yearly graph shows you the long-time trend, eg to see how storage usage has evolved and when you should buy a new hard drive.



16. Backup

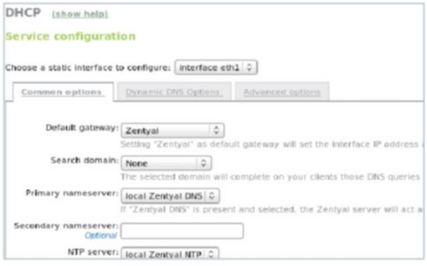
Zentyal can handle backups for you in a lot of ways. You can do this on a local file system (not recommended), using FTP, rsync or SCP, or (if you subscribe to the paid Zentyal Disaster Recovery service) to 'Zentyal Cloud'. Enter a symmetric key for encryption, choose the frequency for a full and for an incremental backup, choose the time when the backup process starts, and configure how many previous full copies you want to keep. Some directories like /var/cache, /tmp, /media, /mnt and so on are excluded by default, and you can change this list in the 'Includes and Excludes' tab.

17. Updates

Of course, the web interface also allows you to keep Zentyal updated and to install new software in the Software Management menu (**Fig 3**). The subpage 'Zentyal Components' gives you access to the installation profiles, and 'System Updates' shows you security and maintenance updates from Ubuntu. If you want security updates to be performed automatically, enable this feature in the Settings subpage.

18. DHCP and DNS

Now that you have most of Zentyal's core configured to your taste, you can begin configuring all sorts of services for your network, of course depending on the components you have installed. We won't delve into the details of them, as the web interface makes most tasks quite easy to configure, but we'll end this tutorial by pointing out some essential services. First and foremost, you can set up Zentyal as your DHCP server to give out IP addresses to your computers and push information like the nameserver and NTP server. If you select 'local Zentyal DNS' as the primary nameserver, Zentyal will act as a caching DNS server, speeding up DNS lookups for your machines.



19. Certification authority

A lot of services can be secured with SSL, but to be able to authenticate your services you need to sign their certificates with a certification authority (CA) certificate. Zentyal allows you to create your own CA certificate: enter your organisation name and details, and the number of days after which your certificate should expire. When you have your CA certificate, you can issue certificates for each of the services you need to authenticate to, such as mail, RADIUS and your web server (these have self-signed certificates by default, but you can replace them with your own). You can always download, revoke or renew a certificate in the web interface.

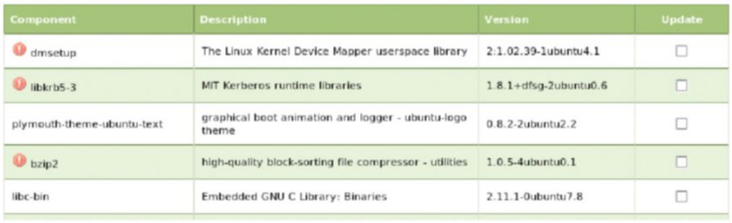
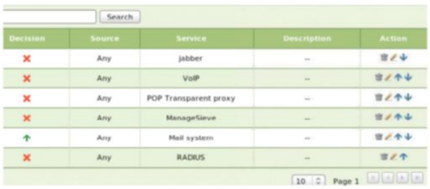


Fig 3 Updates

20. Firewall

The firewall component is one of the more intricate ones. In the 'Packet Filter' submenu, you'll find filtering rules for network traffic in various directions: traffic from the internal network to your Zentyal machine, traffic between different internal networks, traffic to internet, traffic from internet to your Zentyal machine, traffic from internet to your internal network, and so on. You can add and edit firewall rules in each of these categories.



21. Port forwarding

Another submenu of the Firewall menu is 'Port forwarding', which you'll have to use if you run some services on other machines on your internal network that you want to expose to internet. This feature is quite flexible and it offers more than you'll find in most internet routers. You can not only choose the destination IP and port number, but the forwarding rule can also depend on the IP address of the source or the original destination. You can also log new forwarded connections to keep an eye on what's happening.

22. Mail server

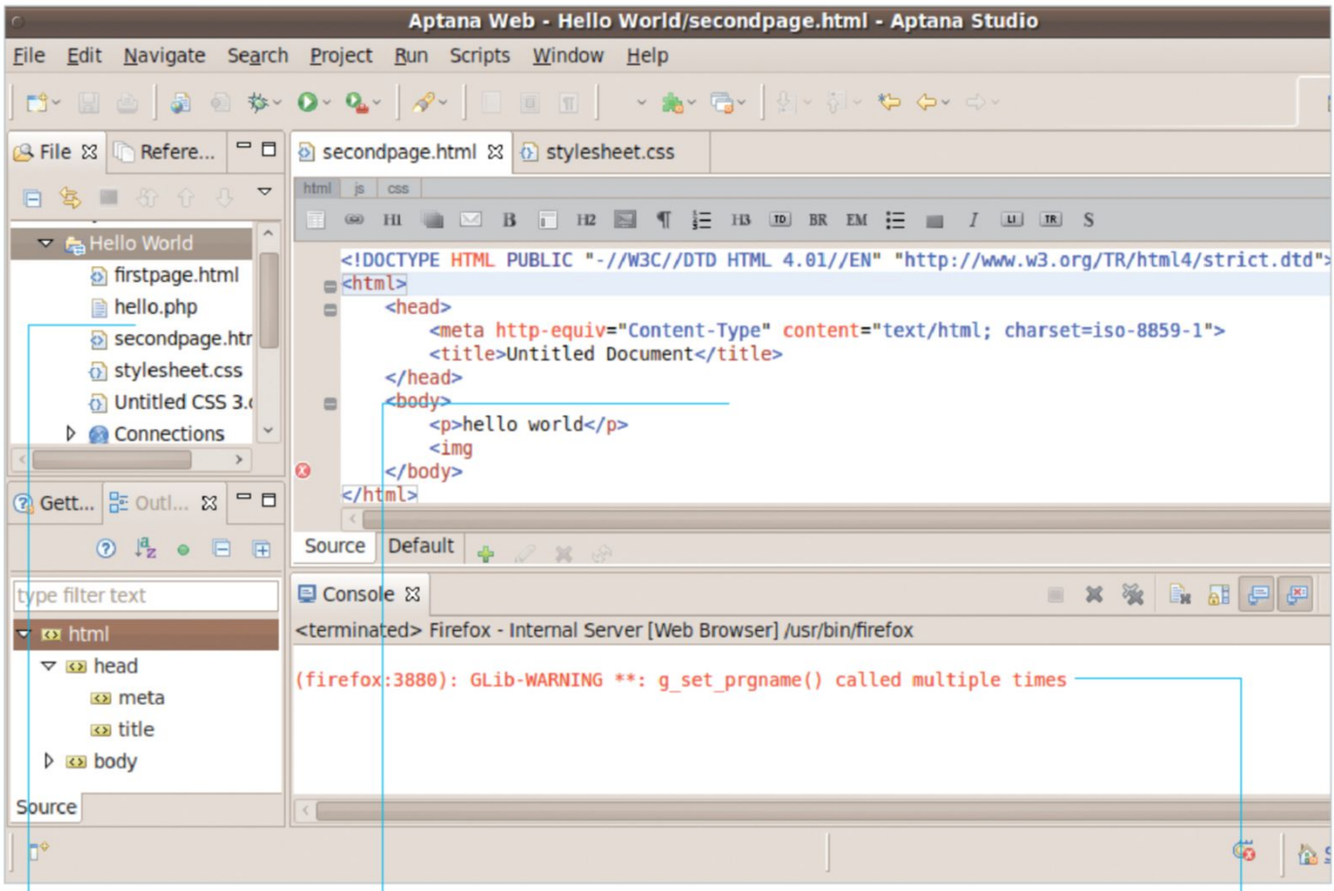
Setting up a mail server is really easy with Zentyal: just install the mail server module and change some options in the Mail menu. You can configure things like the maximum mailbox size, the maximum message size accepted, an expiration period for deleted mails and for spam, and so on. Enable a mail filter in the 'Mail filter options' tab for extra security.

23. Mail filter

If you have the mail filter enabled, all your incoming and outgoing mail will be filtered before it's processed. In the 'Mail Filter' menu you can configure your filter's policy, eg what you want to do with viruses, spam, files with a specific file extension, and bad mail headers. A mail from each of these categories can be passed, rejected, bounced or discarded. You can also edit the file extensions and MIME types to allow and you can tweak the spam threshold.

24. Documentation

Zentyal offers a massive amount of functionality, easily administered by the web interface. You can explore most of these features yourself in the web interface, but if you really want to learn more you should consult Zentyal's extensive online documentation.



The left sidebar contains the list of files, projects and outlines

The code editor has some smart features such as autocomplete and WYSIWYG editing

The console at the bottom shows you any errors that may come up during your development

Build web applications with Aptana Studio

Advisor

Sukrit Dhandhania has spent several years working professionally, implementing several open source tools for companies. During this time he has evaluated, set up and maintained various open source tools for these firms

Aptana Studio is a free, open source cross-platform IDE that supports many languages, such as HTML, CSS, JavaScript and Ruby on Rails

The Aptana Studio open source IDE has many useful features such as FTP/SSH integration, code autocompletion, templates and much more. In this guide we will take a look at how to set up Aptana Studio on your Linux computer and then use it to build and deploy a simple web project.

01 Getting Aptana

It would have been nice to have Aptana available as an easily installable binary such as a DEB or RPM package. However, that does not seem to exist at the moment. That said, the installation process of this wonderful software is not all that difficult. The first step is to download the software from the project's website. Head over to the project's website and download the Linux version of the software – www.aptana.com/studio/download.

02 Install prerequisites

Aptana runs on Java. Aside from Java you also need the XULRunner package. Execute the command '# sudo apt-get install sun-java6-jre sun-java6-plugin xulrunner' on Ubuntu Linux to install the prerequisites (**Fig 1**). If you're using a different distro, refer to its repository.

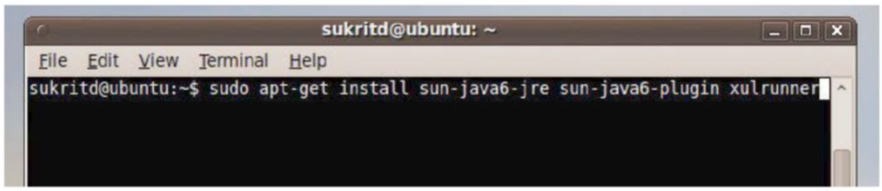
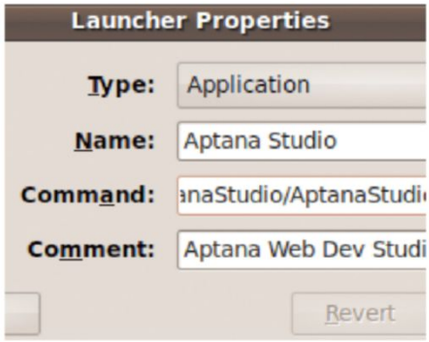


Fig 1 Install prerequisites The installation of the dependencies

03 Complete the Installation

Once Java and the other packages are installed, you can extract the Aptana package you downloaded in step 1 to a location such as /opt/AptanaStudio. Change into the /opt/AptanaStudio directory and launch the application using the command '# ./AptanaStudio'. Aptana Studio should now launch and be ready for use. If you're having problems at this point, refer to the Aptana forum.

■ The Aptana Studio launch window



■ Add a menu item to make it easier for you to launch Aptana Studio

04 Add to menu

One optional step you can perform is

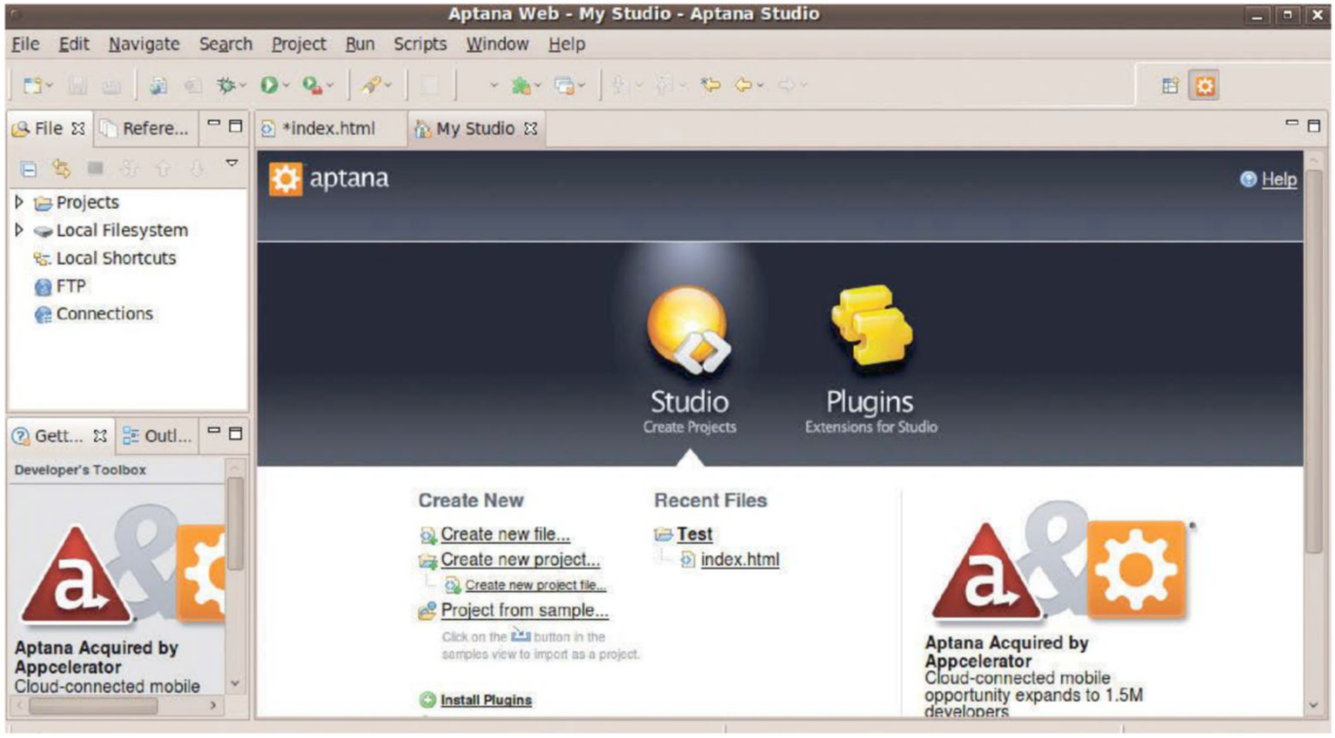


Fig 2 Launch Aptana Studio The main interface of Aptana Studio

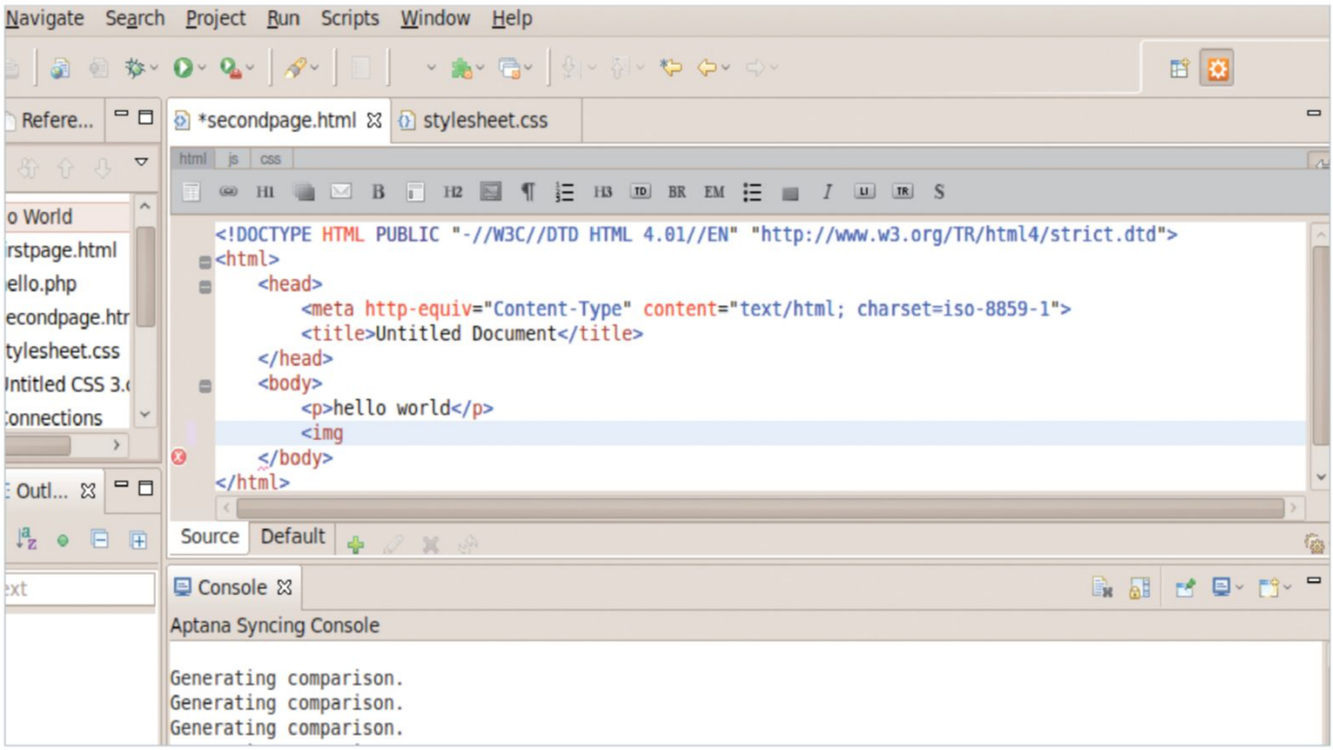


Fig 3 **Create some content** The code editing interface is quite smart, offering autocomplete and other features

to add the Aptana Studio application to the applications menu. To do so, right-click on the menu bar at the top left of your screen and click on the 'Edit Menus' option. Look for the Programming section in the pane that opens and add a 'New Item'. Enter the details and the location of Aptana Studio here, as shown in the screenshot. You should now be able to find the Aptana launcher in Applications> Programming>Aptana Studio.

05 Launch Aptana Studio

After you have set up the menu item, you can began using Aptana Studio. Launch the application from the menu. The first screen (**Fig 2**) will give you an idea of some of the fine features that are part of the package. The left sidebar has all the files, references and outlines, while the main area of the pane is reserved for your code. Let's begin by creating a new project.

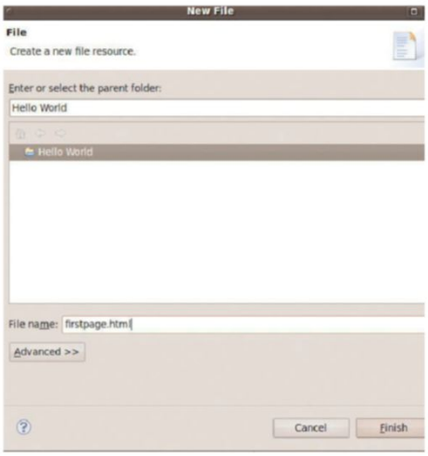
06 Create a new project

Go to File>New>Project to start your project. In the pane that opens, you can choose from a few templates. For now, go with the General>Project option. In the next pane, assign a name such as 'Hello World' to your project. If you want, you can customise the location of the project. In the Referenced Projects box, if you want the new project to reference any of your other projects, check the appropriate projects. Naturally, this option will not be available if this is your first project.

There are a few options for you to choose from when you create a new project

07 Add files

Now that you have created a new project, you can find it in the left sidebar under the Files tab. Expand the Project folder and you should find your new project in there somewhere. Click on File>New>Untitled HTML File. This option will pop up a pane asking you which project you want the new file to be added to, and a name for it.



Add a new file and assign it a name

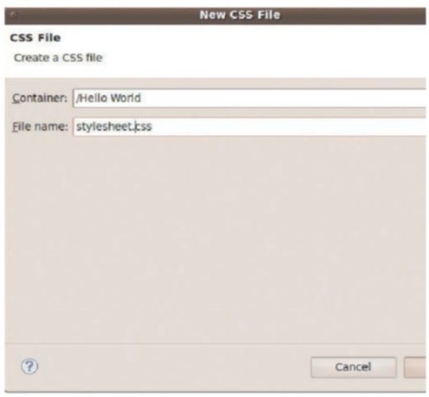
08 Create some content

Finally, you can begin coding. Use the smart code editor to write some HTML code in

the file you just created. Aptana is nice enough to give you a basic HTML template to begin with (**Fig 3**). Enter some content into the page using HTML. Note that you will be entering all of this in the Source tab of the editor. Click on the Default tab to view a preview of your HTML document. You can switch back and forth between these tabs to test your page while coding.

09 Style sheets

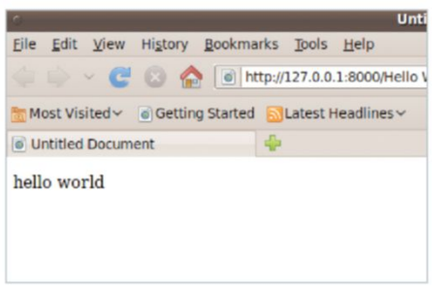
Just like you created an HTML document in the previous step, you can go to File>New>Untitled CSS File to create a new style sheet document for your website. Once the file opens, you can enter your style sheet configuration into it. You will find a few tools to help in the editor itself. There's a couple of colour pickers and a button for entering comments. For some strange reason, when you save the file, Aptana does not offer to make it a part of the project files. You can manually locate the directory containing your project file in your file system and save the file there.



Create a style sheet to make your webpage look a bit nicer

10 Test in a real browser

Once you have made some headway with your web project, you can use a real web browser to test it out. Select the Run option in the menu to launch your web project in a real web browser, most likely Firefox (you can also configure which browser will be used). When done, you can return to Aptana.



Testing it in a real browser gives you an idea of what your users will see

11 The Outline

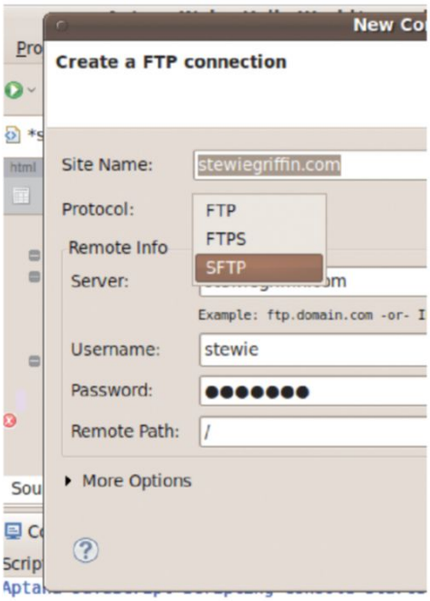
As you are working on your code, you will see the tab titled Outline on the left sidebar becoming populated with entries. This can be of great help as you build more complex applications. When you select a particular entry in the Outline, it highlights the section of code associated with it.

12 Complete your web project

Return to your web project. Create a few HTML pages and make them look nice with a CSS style sheet. Make sure you test things out using the built-in web browser and the web browser installed on your local computer. Save the project to your local hard drive.

13 Add a New FTP/SFTP configuration

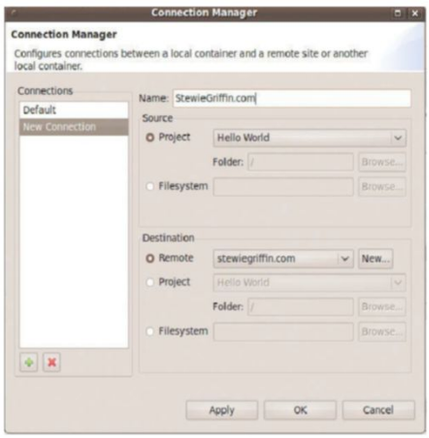
Once the project is complete, or at least at a stage where we can get it online, it is time to set up the FTP/SFTP properties to get that done. On the left sidebar, select the File tab. Scroll down the pane and you will find an option called FTP. Right-click on it to add a new connection. Configure it according to your server settings.



Set up the FTP or SSH connection so you can easily upload your files

14 The Sync Manager

Once you have added this FTP or SFTP connection, return to the File tab and expand your project. You should see a new entry called Connections in there. Right-click on it to view the Sync Manager interface. Here you can set up the details of the connection you wish to use and other details. Now click on the Synchronize button in the File tab. Voila, your first web project should now be online. Although this process may seem overly cumbersome at first, as your projects get more complex you will find that you need many of the options.



Once you configure your server with your project, you can quickly synchronise your web project with your server

"In this guide we will take a look at how to set up Aptana Studio on your Linux computer and then use it to build and deploy a simple web project"

Working with Apostrophe – the CMS with no back end

There are several content management options for users today, both open source and commercial. Let's look at Apostrophe, one such open source CMS which has decided to do things a little differently

Advisor

Sukrit Dhandhania has

spent several years working professionally, implementing different open source tools for companies. During this time he has evaluated, set up and maintained various open source tool for these firms

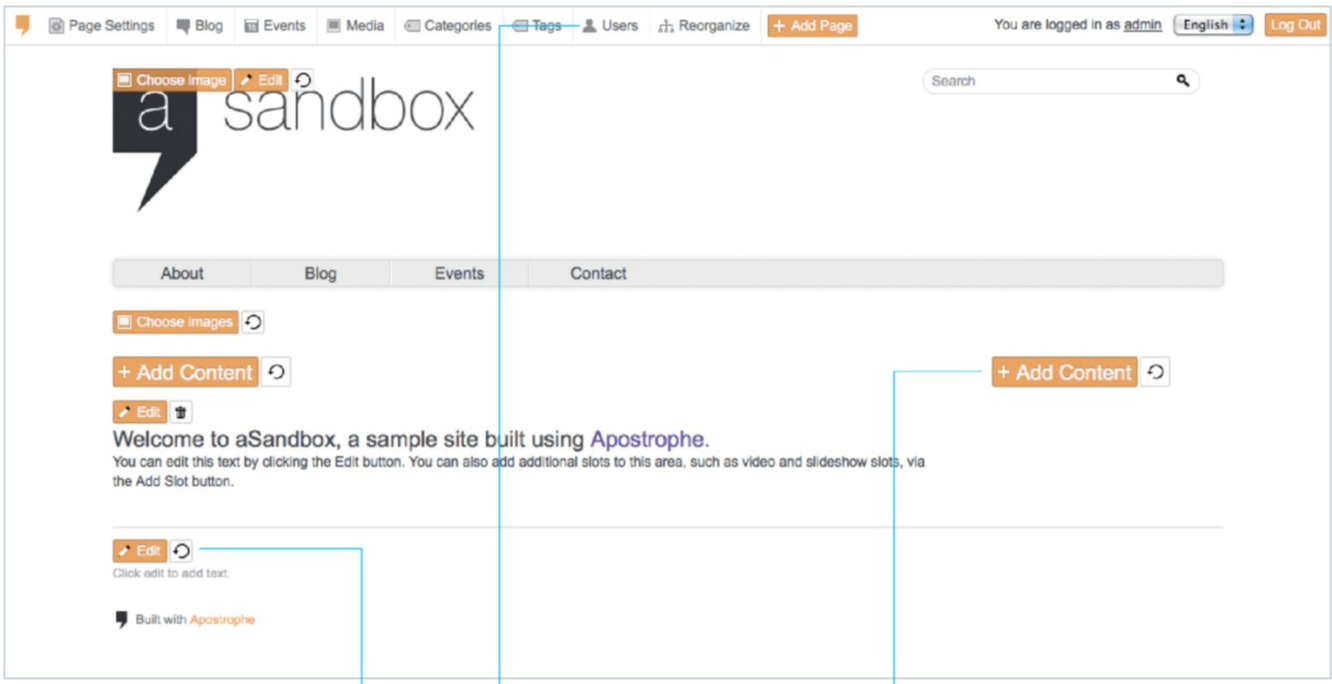
Resources

Apostrophe http://www.apostrophenow.com

If you have used popular open source CMS solutions such as WordPress, Drupal and Joomla, you might have noticed how useful and helpful it can be to use a content management system for your blog or website. Apostrophe is an open source CMS that has gone one step further in making things simpler for its users. The makers of Apostrophe, the wonderful Symfony-based CMS, decided to go without a back end. So you need to enter the content directly into the main page. This makes things much simpler for people who are not used to typing something in the CMS back end and then viewing in the published pages.

01 Choosing versions

There are two ways to get started with Apostrophe. You can either download the Apostrophe plug-in that can integrate with an existing Symfony-based project, or you can get the 'sandbox' version of Apostrophe, which is an independent CMS solution. We'll be working with the latter in this article (**Fig 1**).



Click on the Revision button attached to each post to roll back your posts

At the top you have some basic controls to manage content, media, users etc

The Add Content button allows you to create various types of posts with videos, text and images

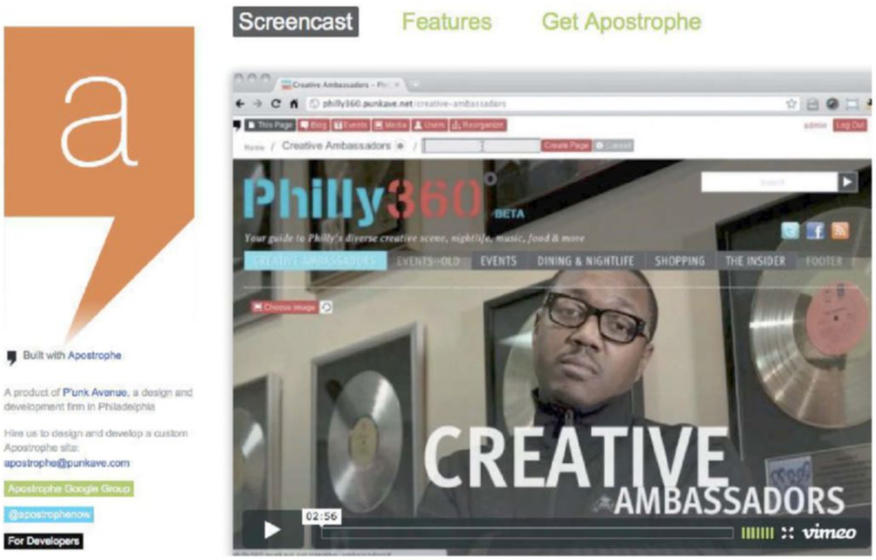


Fig 1 Choosing versions An example of an embedded video in the Apostrophe CMS

> **“Apostrophe has gone one step further in making things simpler for its users”**

02 Set up your server

Apostrophe requires that PHP 5.2.4 or better with the GD libraries and MySQL be installed on the server. If you are using an Ubuntu or Fedora server, these can be installed very easily. You can also install Apostrophe on shared or VPS hosting.

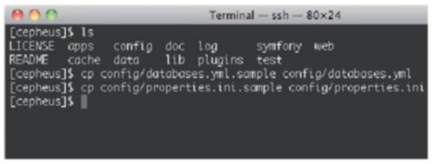
When you get the Apostrophe files, you will find a ‘servercheck.php’ page which verifies these requirements.

03 Get Apostrophe

Assuming your server is ready, we can proceed to getting the Apostrophe files. Again, there are two options. You can either download a tarball of the sandbox project from the site, or you can check out the latest release from the project’s Subversion (SVN) setup. We’ll go with the latter option. Check out the code with the command ‘svn co http://svn.apostrophenow.org/sandboxes/asandbox/branches/1.5 asandbox’. This will obtain the latest release in the 1.5 series.

04 Create config files

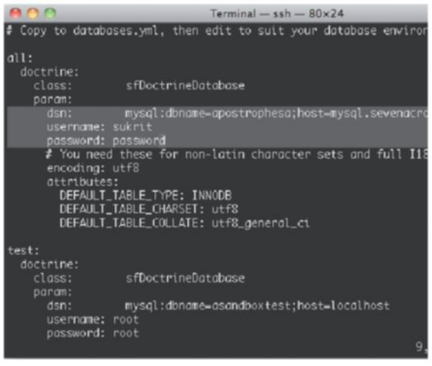
Once the code has been checked out, you should see a folder called ‘asandbox’ on your local computer’s current directory. Copy a couple of sample configuration files by executing the following commands from inside the asandbox directory: ‘# cp config/databases.yml.sample config/databases.yml’, and ‘# cp config/properties.ini.sample config/properties.ini’.



Copy the sample configuration files

05 Set up and configure database

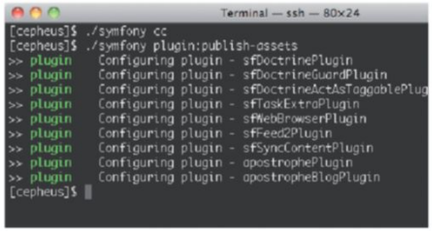
Now create a database for your Apostrophe installation in your local MySQL installation. Edit the database configuration file ‘config/databases.yml’ that you just copied and enter your database credentials in it. You will find two sets of configurations in the file. Set up the one titled ‘all’. Refer to the highlighted section in the screenshot below.



The highlighted section shows an example of the database configuration

06 Compile Apostrophe

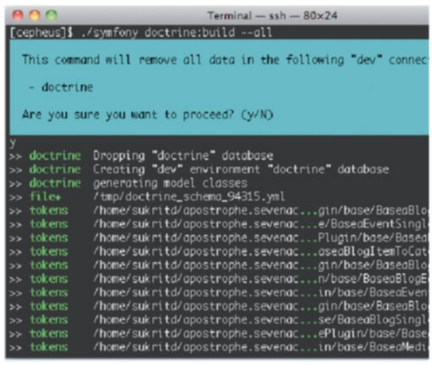
Using the commands ‘# ./symfony cc’ and then ‘# ./symfony plugin:publish-assets’, you will need to compile the application. Check the output to make sure that you do not get any errors.



The output from the compilation process

07 Build Apostrophe

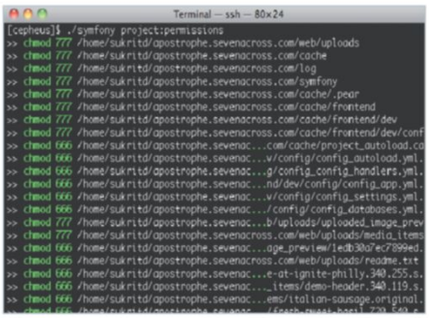
Now you need to build the application. If your configuration is not fine, this bit can be a little tricky. The output of the commands to be run is also quite long, so watch out for errors. Execute the command ‘# ./symfony doctrine:build --all’. If this goes through okay, execute ‘# ./symfony doctrine:data-load’. This last step will set up and load the default data and configurations into the database.



Be careful while building the application, and keep a lookout for errors in the output

08 Load demo data

This step is optional. If you want to load the demo data into the CMS, execute the command '# ./symfony apostrophe:demo-fixtures'. This step can take a bit of time, so be patient. Once that's complete, fix the file permissions across the application by executing '# ./symfony project:permissions'.



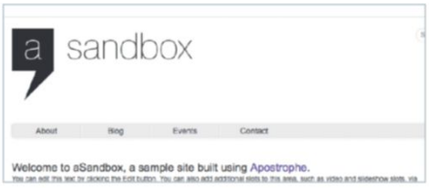
Getting the permissions right is an important part of the installation

09 Configure web server

The last step you need to execute is to make sure that the domain or subdomain being used for this application is set to use the directory 'asandbox/web' as its document root, as opposed to the parent directory 'asandbox'. Without this setting, the application will not work. If you are using a shared hosting setup, you should be able to make this change from the control panel.

10 Houston, we have lift-off

You have wrapped up the installation. So launch you web browser and type in the domain for your Apostrophe installation. Assuming all went well in your installation, you should see something like what's shown in the screenshot below appear in your web browser window. The page looks quite simple, but it's a good start. Now we'll go about looking at the various changes that we can make to its configuration and appearance.



The Apostrophe installation's homepage

11 Log In

After we finished installing Apostrophe for the first time, we spent a long time trying to figure out how to log in. It turns out the default username and password are 'admin' and 'demo'. So hit the Login button at the top-right corner of the homepage and log in so you can start playing with Apostrophe.

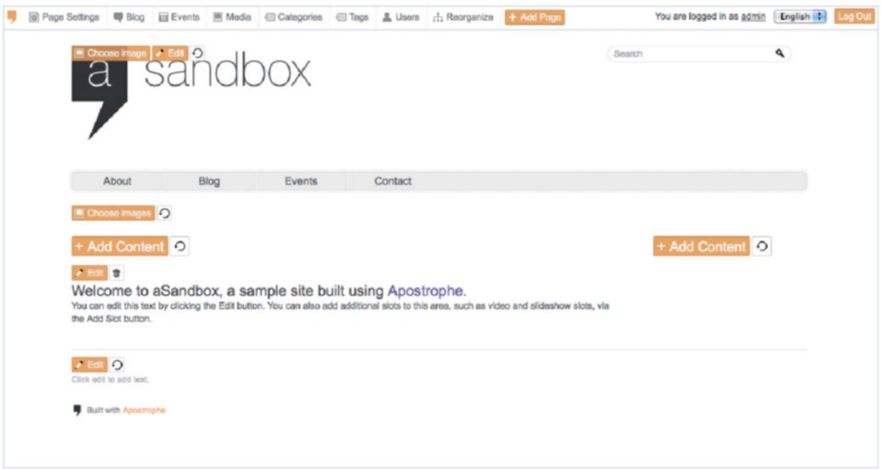
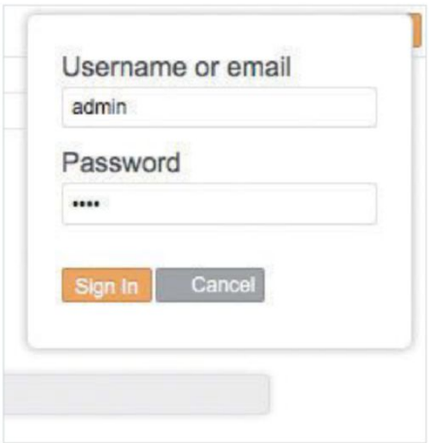


Fig 2 Building blocks The interface will give you an idea of how different Apostrophe is



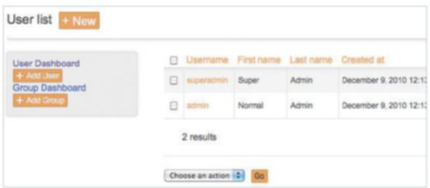
Log in with the authentication of 'admin' and 'demo'

12 Building blocks

When you are logged in, you get a very different view of things than you would if you logged into your WordPress or Drupal blog. You see the scaffolding behind your website (**Fig 2**). Apostrophe has a pretty WYSIWYG approach to creating and managing your content. So if you want to change the title image or link, just click on the 'Choose Image' or 'Edit' link next to it.

13 Users

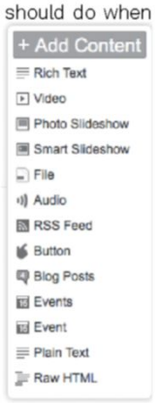
The first thing to do when you log into Apostrophe should be to change the default password to something a bit safer. Click on the Users tab at the top of the page to edit the user's account info. Here you can change a user's permission level, her password, manage groups, and handle a whole bunch of user- and permission-related settings.



Changing the password of the default users is a very good idea

14 Create content

The second thing you should do when you have just installed a CMS is to add some content. So hit the homepage and click on the '+ Add Content' link in the middle of the page. You will be presented with a long list of types of content that you can create. You can create textual content, add videos, create a photo slideshow, upload a file, have audio and so on.



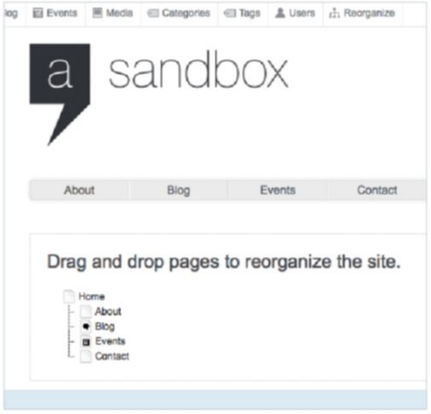
There are many types of content that you can create, like text, video, images etc

15 Picture slideshow

Adding textual content is pretty straightforward. So we'll skip that. The image slideshow, though, is very nicely implemented in Apostrophe. Click on the Add Content button and pick the Photo Slideshow option. Then click on the Choose Images button. You will be taken to the media page. Now upload a few images and then hit Upload Media (**Fig 3**). You will then need to select the file you want in the slideshow and then click on the Save Selection button. Your slideshow should now be active.

16 Organising your content

As you add more posts, you will find it necessary to better handle your content. There are a few different tools in Apostrophe that help you organise your content a bit better. You have five useful buttons at the top – Blog, Media, Categories, Tags, and Reorganize. Using these options you can organise and reorganise your pages, photos and videos, and your blog posts.



Reorganising the pages is quite simple. Just drag and drop pages

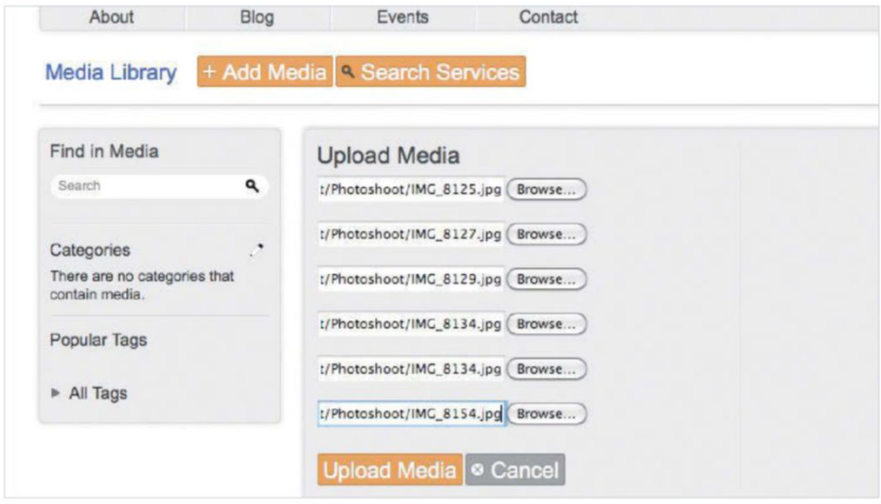
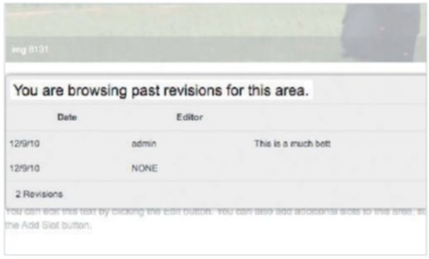


Fig 3 Picture slideshow You can upload several files together – ideal when doing a slideshow

17 Revisions

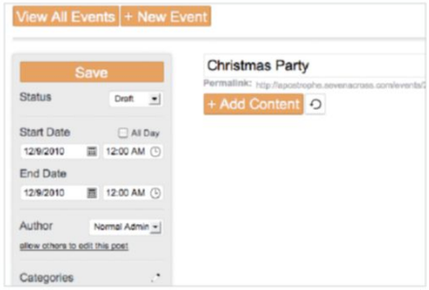
One great feature that is embedded at the top of each post you make is a an icon with a circular arrow on it. Click on this button to view the different versions of the post. You can view the changes you made, roll back and so on. A very useful feature.



Easily view changes and revert to older versions of posts. A very handy option

18 Events

Another useful feature is Events. Although Apostrophe seems to be very much lacking on the social plug-ins front, it has a great way for you to publish your events directly onto your blog or website. Click on Events at the top of the page and create a new one. Set the date and other details and then Save it.

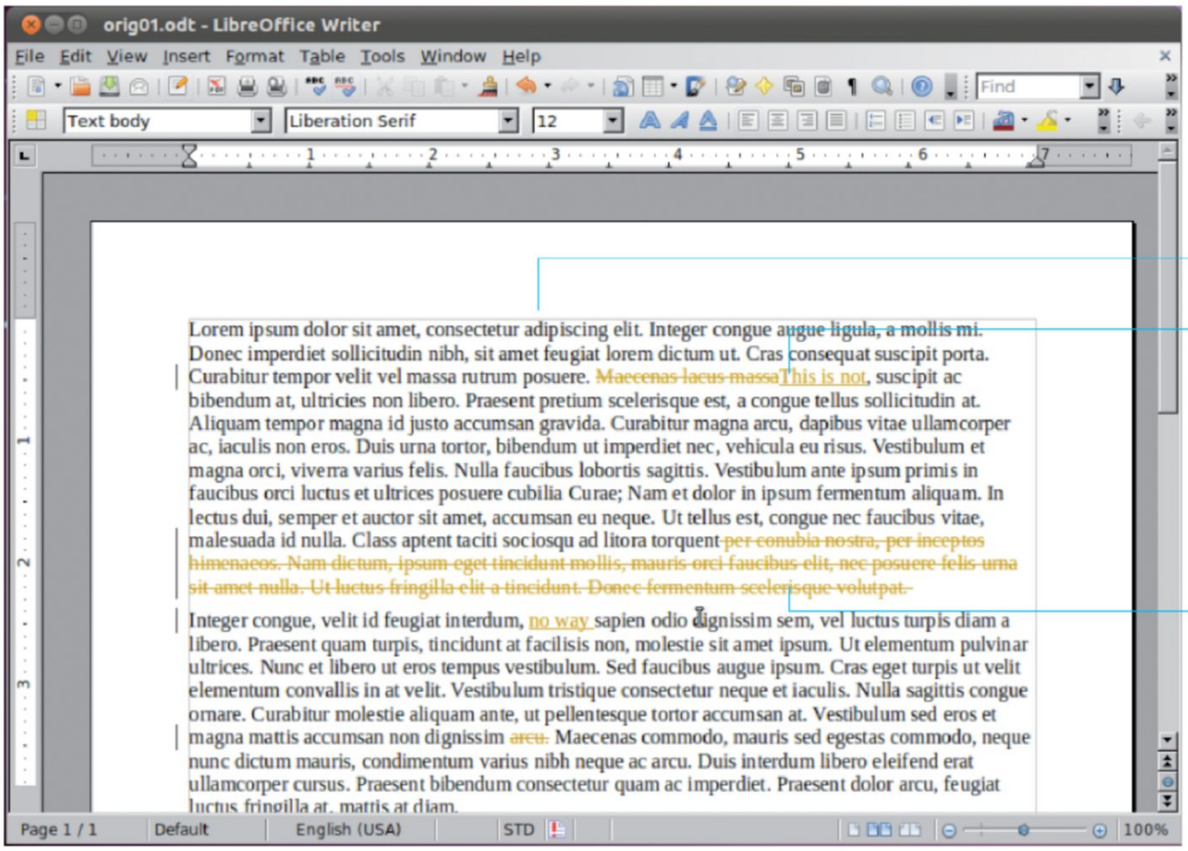


Create and publish events in Apostrophe so can share it with your readers

> **"Apostrophe is a content management system designed for maximum flexibility with a minimal learning curve. The interface is ergonomic, all content editing is performed in context"**
>
> **Source: www.apostrophenow.com**

Conclusion

Apostrophe is a very impressive CMS that is surprisingly easy to get started with. It is packed with some very useful features, and its entirely WYSIWYG design makes for a shallow learning curve. However, there are some things that we didn't quite like about Apostrophe. The first is that there doesn't seem to be a support for themes in it, which is quite disappointing. Another missing feature is the lack of 'social' plug-ins, to be able to share your posts to Facebook and Twitter. All in all, however, it makes for a great alternative to WordPress and Drupal.



All the LibreOffice applications have collaboration features

The highlighted sections are the changes and deleted text

Each author gets her own colour, so you can see different people's corrections

Collaborative document writing with LibreOffice Writer

LibreOffice comes with some useful collaboration features. We take a look at how to make the most of them

Advisor

Sukrit Dhandhania has spent several years working professionally, implementing several open source tools for companies. During this time he has evaluated, set up and maintained various open source tools for these firms

Collaboration has become a very important aspect of our modern work culture. With people working out of different locations, locally and globally, a lot of our software has started to adapt to collaborative development of content. The open source community is not too far behind on this front. LibreOffice is a fine open source office suite for all major OSs. Let's look at how you can use the collaborative features of the LibreOffice word processor, known as Writer. The same features exist on most of the other elements of the office suite.

1. Getting LibreOffice

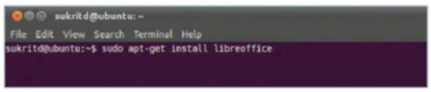
We would like to say that installing LibreOffice is simple and straightforward, just a two-step process. However, it's not. Most of the documentation suggests that you should first remove OpenOffice before installing LibreOffice. You can do that on Ubuntu 10.04 or better using the command '# sudo apt-get purge openoffice*.*'. Now add a repository with '# sudo add-apt-repository ppa:libreoffice/ppa' (**Fig 1**) and update your apt-get database with '# sudo apt-get update'.

Resources

LibreOffice http://www.libreoffice.org

2. Complete installation

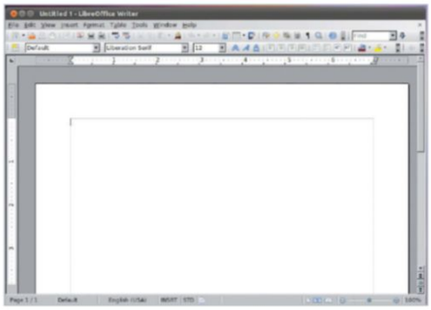
With the repository set up, you need to perform two more steps to complete the installation. Run the installation of LibreOffice using apt-get: '# sudo apt-get install libreoffice'. Then, depending upon whether you are using a GNOME or KDE desktop, you need to execute either '# sudo apt-get install libreoffice-gnome' or '# sudo apt-get install libreoffice-kde'. Now you should be able to access the LibreOffice applications from Applications>Office in the main menu.



■ Install LibreOffice using the newly added repository

3. Launch Writer

Now that LibreOffice is installed on your computer, launch Writer from Applications> Office>LibreOffice Writer in the main menu. This is the tool that we will be using to test out the fine collaboration features of LibreOffice. Enter some text and familiarise yourself with the word processor.



■ The LibreOffice Writer

4. How collaboration works here

Now that we have LibreOffice set up and some content fed into Writer, we can begin collaborating with others as we work on this document. However, let's take a brief step back and try to understand what we mean by collaboration when we talk about it in the LibreOffice context. When you are working on a project with colleagues, have to work on an article with your editor, or want to get your teacher's thoughts on your essay, you can use the Writer collaboration features. Share your document with them; the changes or corrections they make will be recorded and displayed in a different colour. Once you get the document back, you can choose to accept or reject the changes they made. Let's take a look at how all this happens.

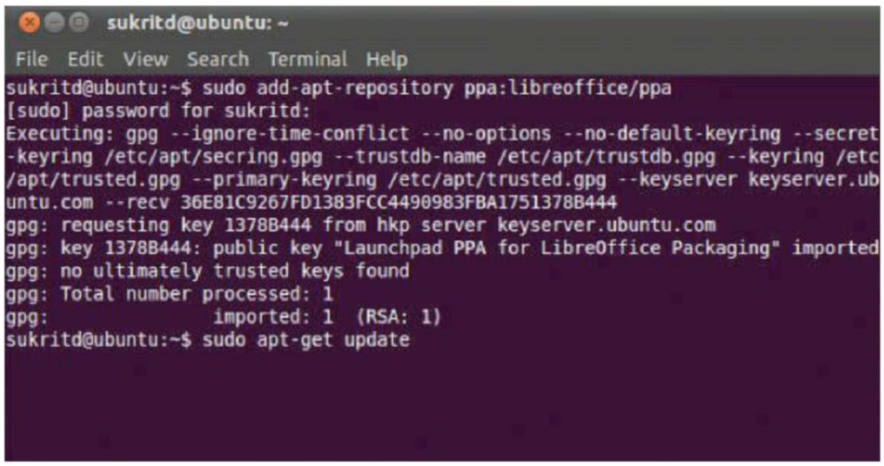
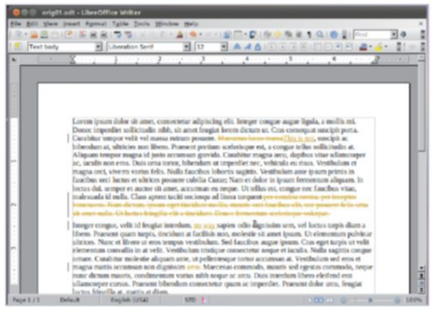


Fig 1 Add a new repository using the Terminal window



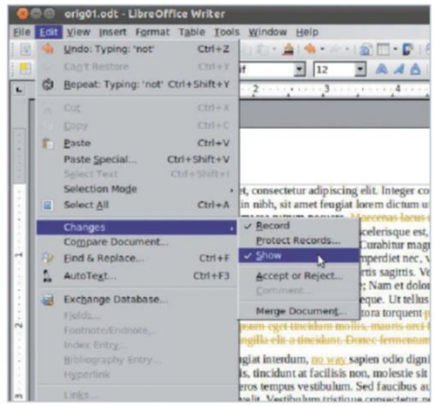
■ Changes are shown in a different colour

5. Recording changes

In a real scenario you would send the document you were working on to your colleague or editor. However, to be practical we'll just save it and open it again. So save the Writer document you started in a previous step somewhere on your computer as something like 'Document-V1'. We will save each version after this with an incremental value. Open the document and go to Edit>Changes and check the Record option.

6. Make your changes

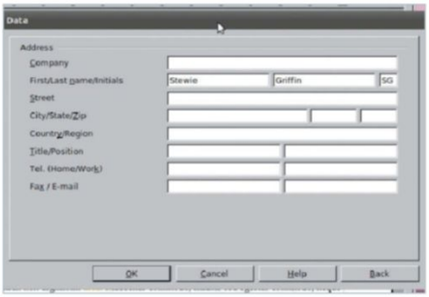
In the Edit>Changes menu item, you should also make sure that the Show option is checked. Now, if you type anything or delete any portion of the text, these changes will be recorded as a new layer. It will be displayed with a different colour and will also be underlined. That way you know what corrections have been made by you as opposed to the original text. You can choose to disable the Show option if you like; the changes will still be recorded.



■ Enable the Show option

7. Return to sender

Once you are done making the changes or corrections you want to make, you can save and email back the changes to your colleague or writer. In our case we will save it as 'Document-V2'. The document will now contain two layers of information. The first layer is the original text while the second one will contain the changes the editor made. When you hover the mouse cursor over the highlighted text containing the corrections made by your editor, you will see the name of the author appear. If it shows up as 'Author Unknown', you should go to Tools>Options, expand the LibreOffice option and click on the User Data option. Here you should enter your first and last names.



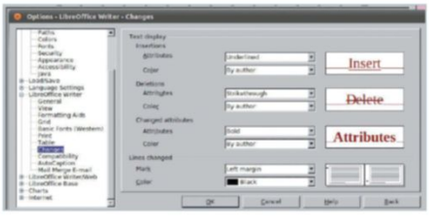
■ If you enter your name, it shows up as the author of the text and in the changes

8. Get more people involved

Just like we did earlier when we had one author creating a document and receive corrections from his editor using LibreOffice's collaboration features, you can collaborate with more than one person in the same way. Each collaborator's changes will be displayed in a different colour.

9. Changing the collaboration preferences

The default formatting of the changed text can be changed if you like. Go to Tools>Options in the LibreOffice Writer menu. Then expand the LibreOffice Writer section and select the Changes option. Here you can view and change the font colour and parameters of the changed text. Change the font size and colour to your taste.



■ Change the colours and other settings for the corrections to your taste

10. Viewing the changes

When you open the document that you saved with the recorded changes, you will still be able to see the different changes that were made by the people that made corrections or additions to your document. These changes will be marked in different colours. At this point you can choose to either accept or reject the changes. Go to Edit>Changes>Accept or Reject in the Writer menu. This will launch a new pane where each suggested change will be listed (**Fig 2**).

11. Accepting the changes

As you scroll down the list of changes, you will see the section of the text affected by the change highlighted in purple. At this point you can choose to either accept or reject individual changes or accept or reject them all. When you act on a suggested change that change will be applied to the text, it will no longer be displayed in a different colour, and the change will disappear from the list in the change pane.

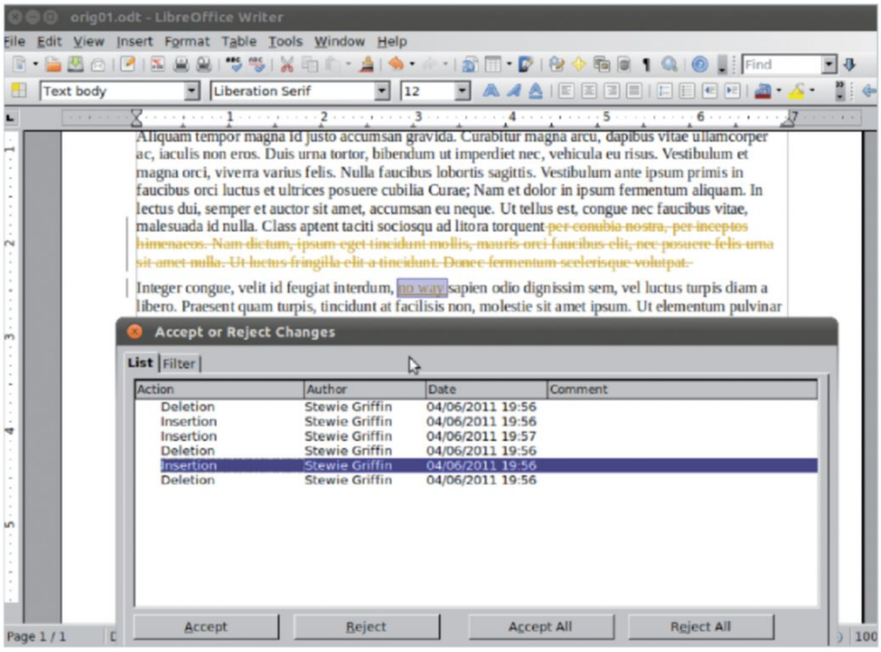


Fig 2 In the 'Accept or Reject' pane, you can view all the changes that were made to your text

12. Filtering changes

Another very useful feature that the LibreOffice team has included is the ability to filter changes. In the pane listing the changes, click on the tab titled Filter. Here you can filter the suggested changes using a number of parameters such as author, date, action and more (**Fig 3**). When you are working on a document with a number of people over a long period of time, this feature can prove very useful.

13. Disable highlighting

When working on the document, you might find it rather annoying to see different portions of the text highlighted in different colours. What you can do then is to go to the Changes menu and uncheck the Show option. Make sure that the Record option is still checked. After you are done with editing the text, you can enable the Show option again and view the changes you made.

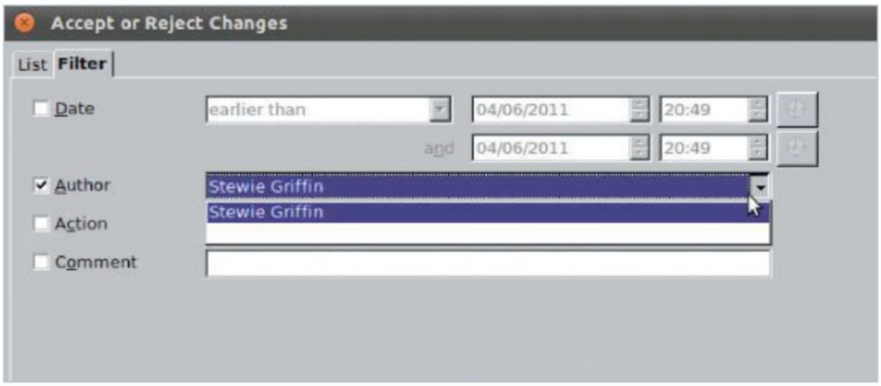
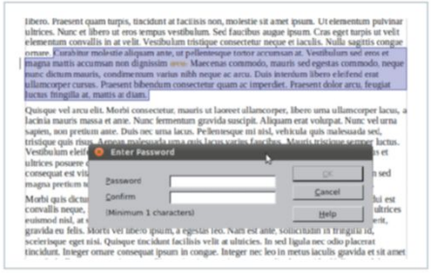


Fig 3 You can filter the changes by author, date, action type and more

14. Protect records

An important feature at your disposal, which is not directly connected to collaboration but could prove to be quite useful, is the Protect Record option in Writer. This feature allows you to select a portion of the text and lock it, thereby preventing any changes to it. This can be particularly useful for sections where you quote someone and don't want your team or editor to change it by mistake. To use this feature, select the text you want to protect then go to Edit>Changes>Protect Records. You will be asked to enter a password to unlock it.



■ Enter a password to protect a section of your text from changes

15. Compare Document

Some people might not be too technically advanced and may not be too willing to use advanced collaborative features in Writer. They may prefer making their changes to your text and sending it back to you. In such a case you can either look for changes and add them to your text. Or you can use the fine Compare Document feature in Writer. Save the new version of your document to some place on your desktop. Go to Edit>Compare Document. This will launch a pane displaying all the differences between the texts. You can pick and choose which you wish to keep and which you want to reject.

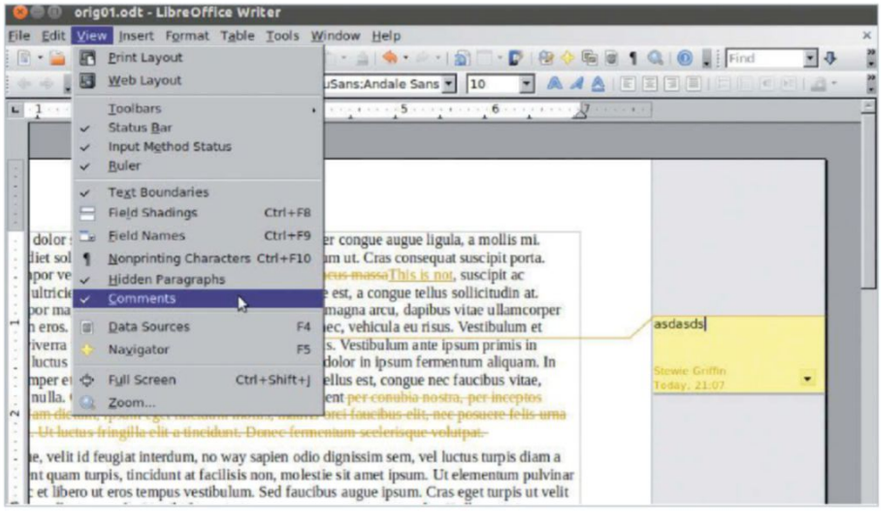
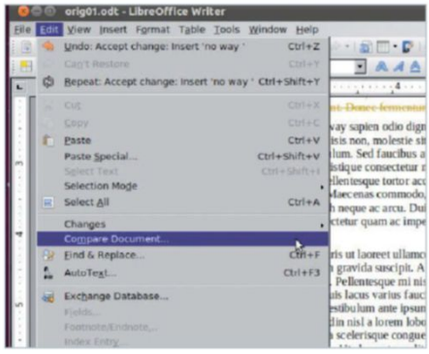


Fig 5 You can enable or disable the comments



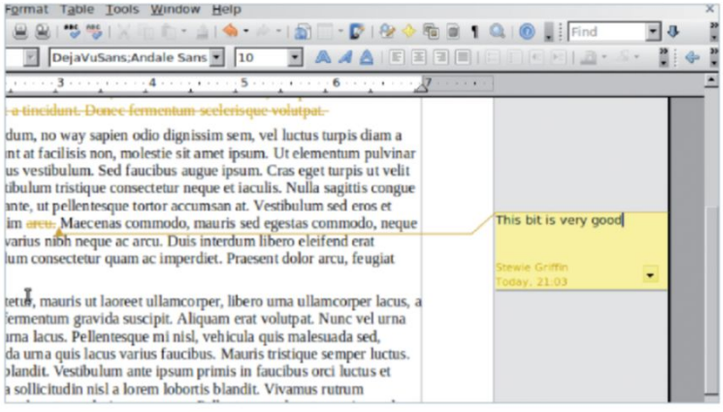
■ Use this feature to compare two versions of the same document

16. Comments

The LibreOffice team have really thought things out while making their implementation of collaboration features. They have included a very nice comments feature. When collaboratively working on a document, sometimes you don't necessarily want to make an alteration to the text itself, but leave a note for your team or the original author (**Fig 4**). In such a case you will find the Comments feature rather useful.

> "You can collaborate with more than one person in the same way"

Fig 4 Each person in the team can leave notes and comments for others

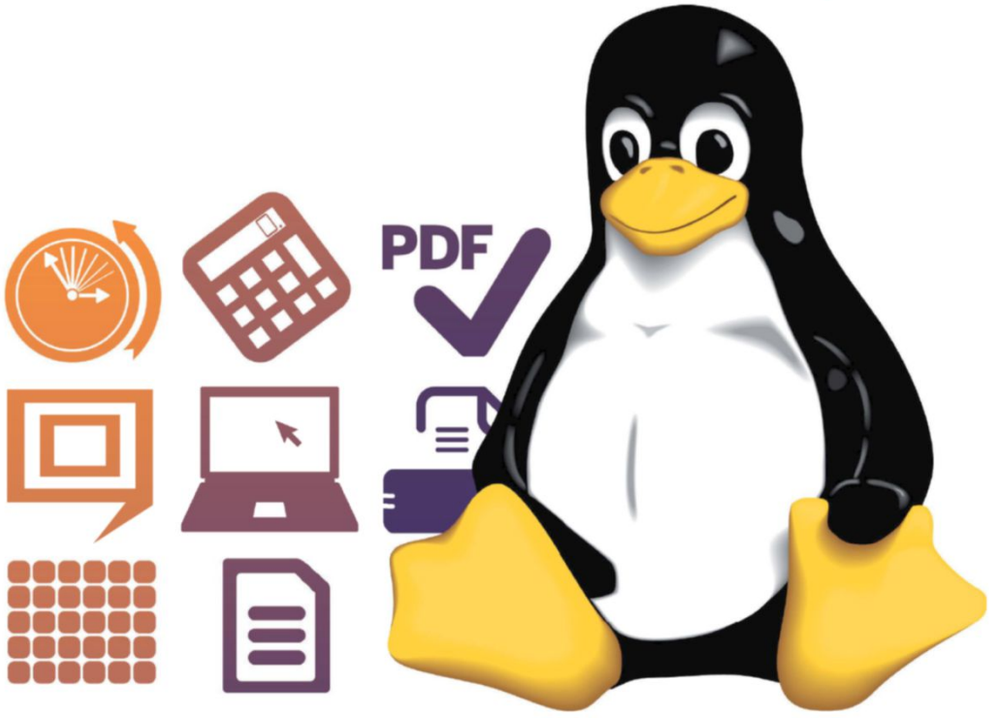


17. Adding a comment

To add a comment to the text, take the cursor to the part of the text where you want to leave the comment and go to Insert>Comment. Alternatively, you can use the Ctrl+Alt+N keyboard shortcut if you prefer. A comment box will then pop up next to the text area. Here you can enter your comment.

18. Hide comments

When reading or editing the text, you may find it annoying to see yellow or blue comment boxes floating on your screen. In such a case you can hide them all by going to View in the Writer menu and unchecking Comments (**Fig 5**). Check it to display all the notes and comments again.



Work more productively with Linux

Desktop Linux has now reached a stage where you'd have to be a bit mad not to evaluate it for office productivity applications. Michael Reed investigates your best options

For some, Linux runs the server, and for others it is a general-purpose desktop. Increasingly, people are finding it to be the ideal business desktop. When weighing the pros and cons of different approaches, bear in mind that Linux paired with FOSS software has some built-in advantages over proprietary solutions. If you want to copy it around the office or pass it on to a friend, no problem. Sometimes you need to tailor a workstation for a specific role – and again, this will prove no problem.

Another advantage of open source over proprietary software in general is that it tends to be unencumbered by lock-in. Most software that was produced purely to make money suffers from that fact that interoperability and backwards compatibility would have meant smaller profits.

Finally, a lot of open source software is simply top class in its own right. It also offers a lot of variety, and it's always worth having a go with a few applications in each field until you find the one that gels with your take on how to get things done.

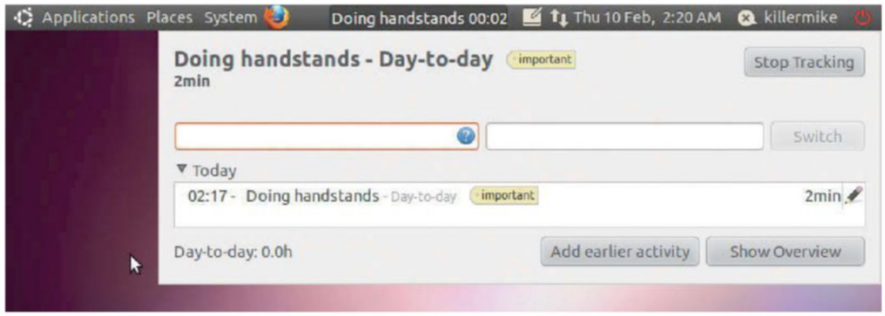
Here we present a handful of office-orientated productivity ideas to inspire you...

Time tracking

If you charge by the hour for your work or you want to find out where the inefficiencies lie in your working methods, a good time tracker is an essential tool. Although there are time-tracking applications that offer a more complex feature set, Hamster (**http://projecthamster.wordpress.com**) offers everything that most people need. It presents itself as a GNOME applet rather than a fully fledged application, which is appropriate because a good user interface is important when it comes to time tracking.

One thing that elevates Hamster above older time-tracking utilities is that it supports tagging of jobs. As with most implementations of tagging, this means that objects can be associated with more than one category at once. As is always the case with tagging, a little bit of extra effort tends to pay off down the line. Let's try an example:

If I start a job that I title 'installing Ubuntu 10.10', I can tag it with 'work, Linux, installing, software'. The next job might be 'design Acme website', which I can tag with 'work, Acme, web design'. Thankfully, once you have begun to work in this way, entering the tags becomes more and more efficient as the Hamster interface allows you to pick and choose from past tags.



Starting a new job in Hamster and remembering to tag

> "One thing that elevates Hamster above older time-tracking utilities is that it supports tagging of jobs"

A small investment of time when entering the tags soon paints an honest picture of how you spend your time. For example, you may discover that you spend a disproportionate amount of time in an activity that doesn't bring in much money.

If you spend a fair amount of time each month just managing your accounts, for example, you might discover that it would be more economic to outsource it to an accountant. Perhaps you'll discover that you spend too much of your time travelling for work doesn't bring in much money? This is the sort of analysis that tagging with time tracking makes possible.

Creating a screencast

Sometimes you have to explain how to do something on the computer, to lots of people. If so, why not make a screencast, an instructional video that you narrate? There are a few GUI tools designed to do this, but they tend to be a bit unreliable. Fortunately, there is a Linux solution that works from the command line – it's called RecordMyDesktop.

Once you have installed the package, which is in the repository of most distributions, you can run it by typing:

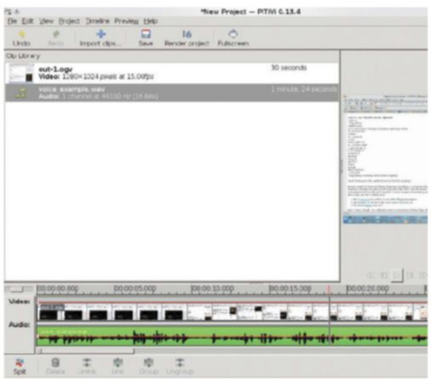
```
recordmydesktop screencast1.ogv
```

This records video of the entire screen at its native resolution. When you've captured the footage that you need, hit Ctrl+C and RecordMyDesktop will convert the captured videofile into an OGV file. The program can be persuaded to convert OGG video files while recording, using the '--on-the-fly-encoding' command-line switch, but this is obviously even more processor intensive. See the utility's man page for a full list of all switches.

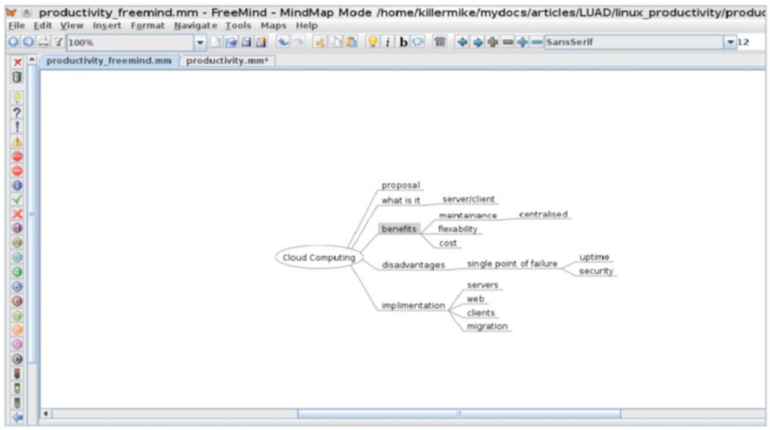
Recording the narration can be done at the time of footage capture or afterwards. The above command will record any audio that enters the system through the microphone port on your computer. In fact, the best approach is often to record something rough while you are carrying out the actions and then replace that with a better version afterwards.

You can use a video editor such as Kdenlive (**www.kdenlive.org**) for final trimming and to add a vocal track. If you want to record a new voice track and your preferred video editor doesn't support capture, consider using the Audacity (**http://audacity.sourceforge.net**) audio editor. You can then recombine video and audio in the video editor, as for most applications, the narration doesn't have to be exactly matched.

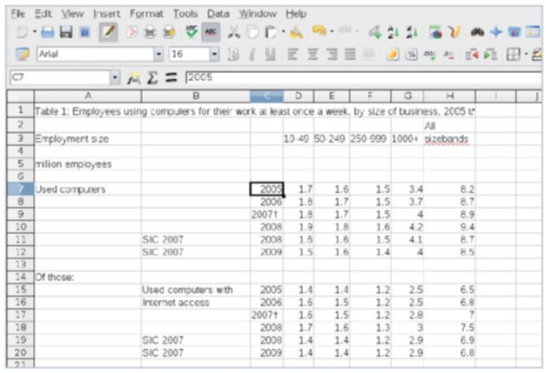
Take our tip: however you decide to work, unless you are experienced with making recordings of your voice, you're almost certainly going to be nervous the first time you try it. Don't be put off though – it's the same for everyone until they've had a few run-throughs. Oh, and no one likes the sound of their own voice when they first hear it played back to them. Once you've got used to it, you'll probably be able to put together a quick screencast tutorial in about half an hour.



Recutting a screencast and adding a new vocal track in Kdenlive



A simple Freemind mind map is a great way of making notes



Conditional formatting makes a spreadsheet easier to work with and more attractive

Note taking

Freemind is a cross-platform mind-mapping application. It uses nodes joined by tree-like branches. Mind mappers are an excellent tool if you like to begin with a disorganised brainstorming session followed by a gradual refinement of the ideas. It can really help you to plan things like essays, reports and business communications.

For example, let's say you want to write a report about employing cloud computing within your organisation. When making a proposal about migrating to cloud computing, you first have to summarise what it is. So, add a node called 'what is it?' to the map. You're going to have to be able to explain what the benefits and drawbacks of cloud computing are, so add nodes called 'advantages' and 'disadvantages' to the map. You should mention the single-point-of-failure issue that affects cloud computing. So, add that to the disadvantages node. Another aspect that you're going to have to carry out is explaining how you intend to implement cloud computing. Make another node for that, and then hang some more nodes off the side of it as they occur to you.

Mind mapping allows a lot of freedom, but a document such as a report tends to be linear. For this reason, prune and rearrange the map so that it is as vertically stacked as possible. Remember that each node is an idea, and if you can't fit the idea into the stack, you can't fit it into the written document. Once you have the plan in place, write it up and amaze your colleagues with your clarity of thought.

Conditional formatting in Calc

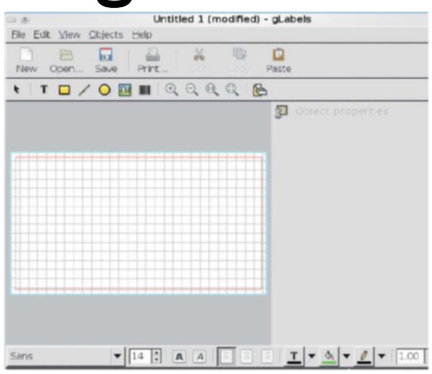
Using colour, or at least shading of some kind, doesn't just improve the attractiveness of a spreadsheet. It also makes it easier to work with, easier to read and, importantly, it allows you to pack more information into the same area. Fortunately, Calc – part of the LibreOffice/OpenOffice suite – has a feature called conditional formatting that can be used to automate the use of shading.

For sake of example, let's say you have a dataset of values that range between 1.0 and 4.2. To make things clearer, let's highlight cells that contain a value of 2 or higher. To do this, first highlight the range of cells that you are interested in by dragging the mouse over them. Now select the 'Cells...' dialog from the Format pull-down menu. From here, select the background tab and choose a background colour for the entire range. While the range of cells is still selected go back to the Format pull-down menu and select 'Conditional Formatting...'. This is where you define a set of conditions that trigger a different formatting style. In this example, use the drop-down menus to specify 'cell value is', 'greater than' and the number 2. Having done this, select the 'New style...' option from this dialog. Then select the Background tab, but this time select a colour that contrasts with the overall background colour that you selected. Click on OK and the changes will take effect.

Label printing with gLabels

These days, the age-old dream of the paperless office is more or less with us. However, sometimes, folders full of printed reports, parcels and other stubbornly tangible items have to be labelled. Enter gLabels. It's a label printing tool that can handle pretty much any type of label you can think of, and probably some that you hadn't.

The program has two main windows. The first is the label designer, in which you can add text, lines and boxes to the label. The second window is the template window and it contains literally hundreds of starting points for a standard label and business card designs.



Enter the text (or address line numbers from a document merge) and add logos etc in the label designer screen of gLabels

Portable office

If you often move between workplaces or work on more than one computer, a portable Linux installation can be a godsend. Nearly all of the major distributions can be installed onto removable media such as a USB memory stick. By using Linux in this way, you also gain the dual advantage of not having to disturb the files on the native PC, and at the same time you won't be affected by whatever software (invited or not) already lives on that computer.

If you've not tried this approach before, you may think that it involves some hefty compromises, but if you're using it for office apps, you don't have to give up much. If people were honest about what applications they need in order to do their work, they would probably find that they only require a word processor, a web browser, maybe a file manager, and a handful of small applications such as the ones we've discussed here. Once you've tried working like this, you might begin to wonder if you need a dedicated, hard-disk-based work setup at all. Another point in favour of a nomadic setup like this is that if your computer blows up one morning, you simply find another computer, plug in your flash drive and then get back to work.

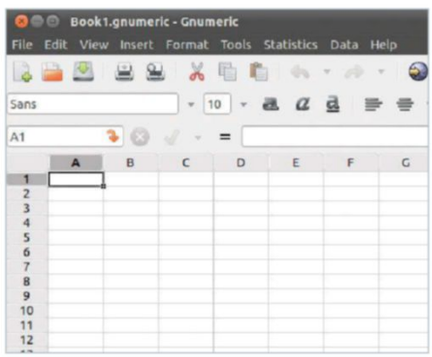
Amazingly, you could carry this this fully featured desktop around in your pocket

Puppy Linux (**http://puppylinux.org**) is a good choice for this sort of setup. It's fairly lean, it's extensible and it comes with some tools to fettle awkward or archaic hardware. Since version 5, Puppy is based upon the Ubuntu packages, so you're highly unlikely be stuck for appropriate software. Installing it onto a memory stick is a cinch. Begin by burning the Puppy Linux ISO onto an CD-R, then boot off this disc in the normal way. Once you've reached the Puppy Linux desktop, double-click on Install and select the menu options for a USB installation. As well as a complete desktop system, such a Puppy-based setup is useful for recovery of a non-booting PC.

The GNOME office suite

The so-called GNOME Office Suite (http://live.gnome.org/GnomeOffice) is actually a loose collection of productivity applications that can be optionally employed together, and they tend to get overlooked in favour of the mighty LibreOffice. It's a shame as they are some of the most polished Linux applications available, and they might even offer some advantages over the more famous suite. Although some might regard the look of these applications as old-fashioned, people who don't live life on the cutting edge often welcome the traditional layout that they sport. They're also great for a lightweight installation, as LibreOffice is something of a heavyweight.

AbiWord is the word processing component. A lot of people are attracted to it because it looks similar to word processors that were popular ten or 15 years ago. This means no ribbons and no flower fairies popping up with words of advice, and a lot of people like it that way. However, AbiWord itself is a modern application that just happens to have a classic approach to the way its user interface works. It fits well into typical business situations as it works with most standard file formats.



The Gnumeric spreadsheet program is dependable and well documented

Gnumeric is a fully featured spreadsheet program. It offers everything you'd expect from a business-class app and, as it's been around for ages, it's dependable and well documented. It also has a good set of graphing features. Speaking of graphics, the GNOME office suite offers vector drawing package Inkscape, arguably the most comprehensive open source illustration app. Dia is a powerful program for quickly putting together diagrams that reuse graphical elements. **Adobe's PDF format is now the de facto standard for portable documents.** Support on Linux is good, thanks largely to the fact that the format is well documented and open. Luckily, almost every Linux application that deals with documents can output to PDF, either through a dedicated facility or by simply printing to a virtual device that produces a PDF file.

Working with PDF files

Software is also available to edit PDFs. As the name suggests, PDFedit (**http://sourceforge.net/projects/pdfedit/**) is a tool for making alterations to existing PDF files. The main use for this facility in an office situation is to annotate a PDF file with graphics and text. It's also handy to be able to fill in a form entirely from within the computer instead of printing it out first.

Run Linux

Join Kunal Deo on a fast-paced thrill-ride showing us how to install Linux on a variety of incredibly common household gadgets...

Linux has always been thought of as the hacker's go-to operating system. In the open source world, the best code is the code you don't have to write. Imagine you've just hacked an awesome gadget and that gadget is now able to run your code. What would you do with it? Would you write every piece of software yourself from scratch, or would you try to find a way for it to run a selection of tried and tested software already out in the wild? Unless you're every crazier than us, it would be the latter and the latter means Linux. With Linux installed, the sky is the limit.

Linux is the only OS that supports all the main hardware platforms (x86, ARM, MIPS, PowerPC and so on) and if a dedicated group find a platform that fits their needs but doesn't yet support Linux, they just get it working.

Read on to find out how you can install Linux on some of the most unexpected household gadgets. This list is by no means definitive, but it provides a great introduction to installing Linux on products that you've probably got lying around the house...

iPod classic
Add extra audio and video codec support and even play retro games on it

Nintendo DS
This affordable handheld games system can run DSLinux, enabling you to practise UNIX/Linux commands

HTC Desire HD
Install Ubuntu 10.10 and run almost any Ubuntu application on your smartphone

anywhere

Warning

1. Information presented here is for educational purposes only. The article author and Imagine Publishing Ltd do not condone piracy and strongly advise our readers against it.

2. While the information presented here has been tested, we take no responsibility if something goes wrong or your device becomes unusable.

3. Modifying system software on your device may render your device's warranty void. Check with your vendor for more information before proceeding. We will not be held liable for any damage caused.

4. Linked websites are not connected to Imagine Publishing Ltd. We can not take any responsibility for their content.

Samsung TV

Add extra features to your internet-enabled TV and even play emulated games on it

Nintendo Wii

Several Linux distros are available for what is one of the most-hacked videogames consoles ever made

Router

Unlock the Linux distribution in your network/Wi-Fi router to open up a whole new set of possibilities

Nintendo Wii

The Nintendo Wii is an iconic device in the videogame world. As well as being the only console to make a profit from hardware sales (in comparison to Microsoft Xbox 360 and Sony PlayStation 3, which are being sold at a loss), it is the bestselling console of the current generation. There are lot of reasons behind its success, such as affordability and its unique touch controls. Because of its popularity, the Wii has also become one of the most hacked videogame consoles ever made. Multiple Linux distributions are available for it, thanks to the already existing (and insanely active) GameCube hacking community.

Hardware required

Nintendo Wii with all the standard accessories.
A USB keyboard and USB storage device.
4GB or more SD card.

Firmware version required

In Wii terms, we are talking about Wii System Menu. For PAL, NTSC-U, NTSC-J consoles you can use Wii System Menu ranging from 3.0 to 4.3. For a Korean Wii you will need Wii System Menu ranging from 3.0 to 4.1. Different hacks are required to make your Wii homebrew capable for each system.

Packages required

Installer Kernel Package, available from **http://downloads.sf.net/gc-linux/whiite-linux-installer-BETA1.tar.bz2** and a compatible bundle package, available from **http://downloads.sf.net/gc-linux/whiite-linux-bundle-0.1_2.6.27b.tar**. More version can be found in sourceforge.net's repository.

Installing Linux

1 Install the Homebrew Channel. Depending on your Wii version, you will need to either use a game exploit or banner bomb. See **http://wiibrew.org/wiki/Homebrew_setup** for details.

2 Once you have the Homebrew Channel up and running, you can install Linux on it. There are several distributions available for Wii (**http://wiibrew.org/wiki/Wii-Linux/Distros**), but we recommend using White Linux, as it gets more updates and has a nice installer which is easy to use.

3 Install White Linux using White Linux Installer. See **www.gc-linux.org/wiki/WL:whiite-linux-installer** for details.

White Linux Installer in the Homebrew Channel

Now what?

You can use your Wii as a typical Linux desktop. Since it has only 24MB of RAM, only a text-based interface is available. You can run various servers (web, SSH etc), play MP3s and use IRC. You can also install various Debian packages using the regular apt-get command.

Risks involved

If you are using a Korean Wii with US firmware, do not update to the latest firmware, as doing so may turn your Wii into a brick.

"The Wii has become one of the most hacked videogame consoles. Multiple Linux distros are available for it"

Nintendo DS

The Nintendo DS is one of the most successful handheld consoles in history. It boasts a large number of game titles. Currently, the Nintendo DS Lite is available at a very low price, making it ideal for hacking. You can install DSLinux on it, which is based on µClinux. Because of its extremely low memory (4MB RAM), you can't expect a blazing graphical Linux, but you will get a nice text-based UI.

Hardware required

Original Nintendo DS or Nintendo DS Lite.
A SLOT-1 cartridge which allows you to run homebrew software stored on the memory card. See Wikipedia for more information on SLOT-1 cards – **http://en.wikipedia.org/wiki/DS_Homebrew#SLOT-1_and_SLOT-2_devices**.
A microSD memory card.

Firmware version required

DS firmware should be compatible with the SLOT-1 card. Check with the vendor of the latter.

Packages required

SLOT-1 firmware. Check the SLOT-1 vendor website to download the relevant firmware.
DSLinux build, available from **http://kineox.free.fr/DS/dslinux.nds**.

HTC Desire HD

HTC's Desire HD was one of the first handsets to bring desktop-like power and a large screen to a smartphone. It has a 1GHz processor, 4.3" screen (supporting 400x800 resolution) and 768MB of RAM. So, as you can see, this Android device packs a punch. Installing Linux will enable you to unleash its raw power. Now you may ask, isn't Android Linux anyway? Well, yes it is, but what we are talking about is installing Linux proper – Ubuntu Linux, to be precise. We will go through the process of installing Ubuntu Linux 10.10 on the Desire HD.

Hardware required

HTC Desire HD smartphone.
microSD Card with a minimum of 3GB space free

Firmware version required

1.7xx stock ROM or any custom ROM.

Packages required

Android custom kernel with LoopBack support. You can download one from **www.multiupload.com/U2ECLTRT7V**.

Ubuntu 10.10 distribution customised for Desire HD and required scripts. You can download it from **http://leedroid.com/file_download/173/ubuntu.rar.torrent**.

An Android Terminal emulator from the Android Market.

Android VNC from the Android Market.

Installing Linux

1. Download Paul O'Brien's VISIONary+ tool (we used version r12), which enables root and removes new HTC protection. Google removed this app from the Market, so you'll have to download it from Paul's website – **http://android.modaco.com/content/htc-desire-hd-desirehd-modaco-com/320722/19-nov-r14-visionary-one-click-root/**.
2. Root your HTC Desire HD device using jkoljo's One-click Radio S-OFF tool. Follow the instructions presented here: **http://forum.xda-developers.com/showthread.php?t=857537**.
3. Install RomManager from the Android Market and do a full backup.
4. Follow the Ubuntu Linux installation instructions from LeeDroid – **http://forum.xda-developers.com/showthread.php?t=963385&highlight=Ubuntu**.

VISIONary+ root tool

Now what?

You now have a desktop-class Linux installed on your smartphone. You can run the fully fledged Firefox web browser for restriction-free web browsing. You can also run almost all standard Ubuntu applications.

Risks involved

Be careful while rooting the phone. When in doubt, always use the one-click tool to do so. If the process is not done properly, you can corrupt the bootloader.

Installing Linux

1. Download and install the firmware for the microSD card.
2. Put the downloaded DSLinux build on the microSD Card and boot from it.
3. If you need additional builds or the source code, visit **http://kineox.free.fr/DS/**.
4. Log in with username 'root' and the password 'uClinux'.

Now what?

You can learn and practise most of the UNIX and Linux commands right on your DS.

Risks involved

Non-specific. Standard risks still apply.

"Because of its extremely low memory (4MB RAM), you can't expect a blazing graphical Linux"

DSLinux in the Homebrew Menu

Samsung LCD /LED TV

Samsung internet-enabled TVs (also known as Internet@TV) are great. They allow you to install applications and use Linux to achieve that. But it is locked. In this section we will learn how unlock the Linux to install our own applications on Samsung TVs. We will be using the SamyGO project to unlock the TV.

Hardware required

A compatible Samsung C Series TV. This includes UExxC6800, UNxxC6900, LExxC650. For a complete list of compatible C Series TVs, see **http://sourceforge.net/apps/mediawiki/samygo/index.php?title=Compatibility_Table_for_C_series_TVs**.

A USB flash drive with 4GB or more free space.

Firmware version required

Most of the stock firmwares are supported. See **http://sourceforge.net/apps/mediawiki/samygo/index.php?title=Compatibility_Table_for_C_series_TVs** for a complete list of supported firmwares.

Packages required

widget.zip can be obtained from **www.multiupload.com/OLQTLJ8G52**

SamyGO Extension Pack for your TV's firmware. This pack is available from **https://sourceforge.net/projects/samygo/files/SamyGO%20Extensions%20Packs/**. You can look up the firmware version by downloading the firmware update from Samsung's support site and reading its filename. For example, T-VALDEUC.

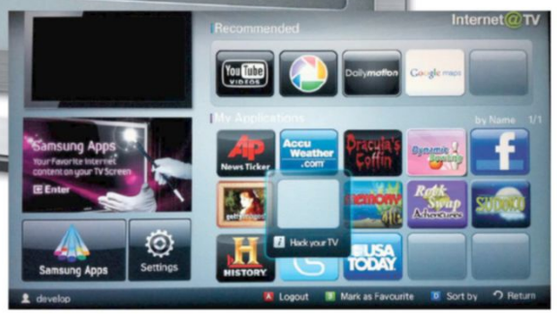
Hacking device

1 To do the hacking using only the Internet@TV, see **http://http://sourceforge.net/apps/mediawiki/samygo/index.php?title=Hacking_C-Series_TV_with_Internet@TV_only**.

2 To do the hacking using the Hotel mode, see **http://sourceforge.net/apps/mediawiki/samygo/index.php?title=Hacking_TV_over_Hotel_mode_(most_C_series_models)**.

Now what?

Your TV is now enabled to do crazy things (we mean in a good way). You can add more features to it (such as Audio Stream Switcher), add servers (such as an FTP server) or even play games by installing emulators.



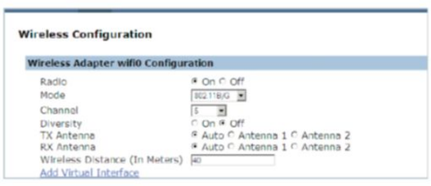
The Hack Your TV app installed in the Samsung Internet Applications

Risks involved

Hidden menus on your TV (like the Hotel menu and Service menu) are a dangerous place for you to experiment. Stick to the instructions and do not do anything else unless you really know what you are doing. Making an unwanted change could turn your large-screen TV into, well, just a large screen.

Network router

Most of us use a network (or Wi-Fi) router on a daily basis, but do not realise that a router is actually a fully functional stripped-down PC with Linux installed in it (for the most part). As mentioned in the case of Samsung TVs, the Linux distributions shipped with these routers are locked down. And unlocking the Linux could open up a whole new set of possibilities. In this section we will be using OpenWrt to unlock a Linux-based network router. OpenWrt is a Linux distribution for embedded devices.



OpenWrt wireless LAN configuration

iPod classic

The iPod classic is the device which revolutionised the music world. Even after being cannibalised by its iPhone-lookalike sibling the iPod touch, it is still popular because of its large storage capacity and low cost. We will be installing the iPodLinux distribution on it. iPodLinux is a µClinux-based Linux distro designed specifically to run on Apple's iPod.

iPod Linux Loader

iPodLinux running Podzilla

Hardware required
An iPod classic model supported by iPodLinux. This includes iPod classic generations 1st to 6th.

Firmware version required
Non-specific.

Packages required
Non-specific. The iPodLinux project recommends you build the installer yourself. So you will need mandatory development tools installed on your Linux distribution.

"iPodLinux gives your iPod classic a new lease of life by adding support for new audio and video codecs"

Installing Linux

1 Build the Installer 2 package for iPodLinux. See **http://ipl.derpapst.eu/wiki/Installer_2** for instructions on how to build Installer 2.

2 Install the iPodLinux distro using Installer 2. For further instructions, see **http://ipl.derpapst.eu/wiki/Installation**.

Now what?
iPodLinux gives your iPod classic a new lease of life by adding support for new audio and video codecs. You can also use your iPod classic as a retro gaming console by installing emulators such as iMAME, iNES etc.

Risks involved
Non-specific. Installer 2 will automatically do a firmware backup for you. You can also use the iTunes Restore function to return the iPod classic to its original condition.

Hardware required
One of the supported routers mentioned at **http://wiki.openwrt.org/toh/start#supported.hardware.-.router.type**.
Depending upon the router model, you may need additional hardware.

Firmware version required
As per the Table of Hardware, available at **http://wiki.openwrt.org/toh/start**.

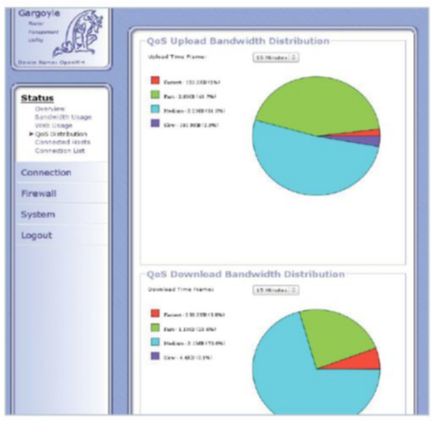
Packages required
Visit **http://wiki.openwrt.org/doc/packages** for information on offical, custom and third-party packages.

Installing OpenWrt
Each router follows its own installation method. Please see the Table of Hardware for the relevant installation method for your router – **http://wiki.openwrt.org/toh/start**.

Now what?
With OpenWrt installed, your router has just become a super-router. You can set up VPN and Netboot, monitor bandwidth usage, set up a database server or a file server... You can even set up PBX software such as the Asterisk on your router now.

"Most of us don't realise that a router is actually a fully functional stripped-down PC with Linux installed in it"

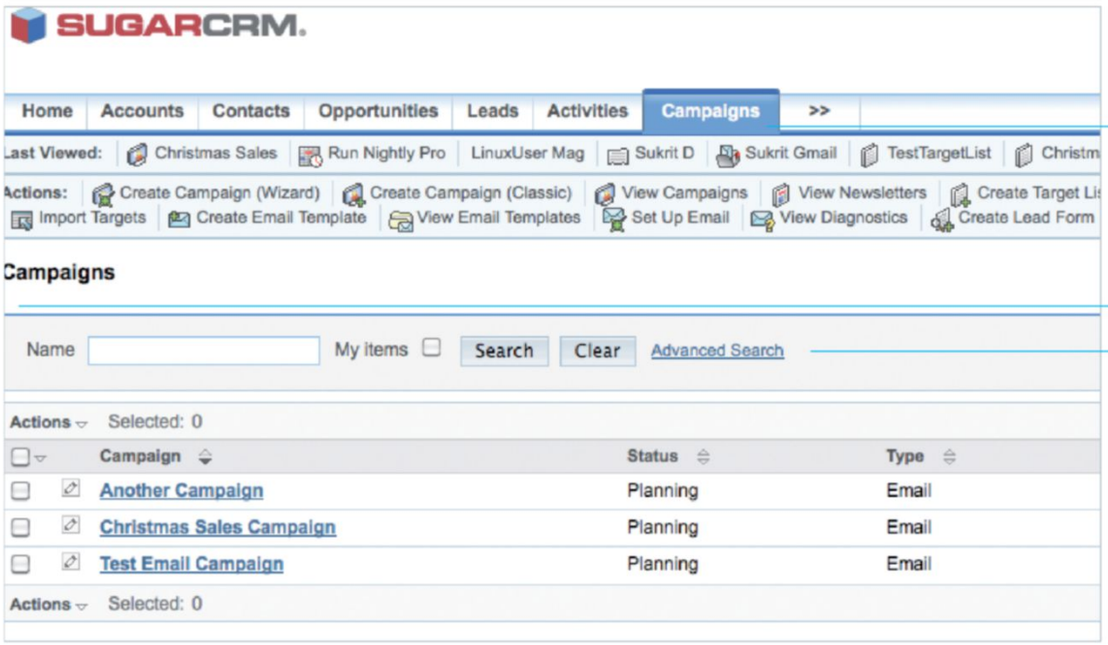
Risks involved
Do not physically open a router unless you have experience in electronics soldering.



OpenWRT QoS pie chart

Create an email campaign with SugarCRM

Create and send an email marketing campaign using SugarCRM – one of the most popular open source CRM solutions available



View the list of campaigns that have been setup, and edit or launch them

View the list of campaigns that have been setup, and edit or launch them

You can access all the settings related to email campaigns such as Target Lists and Campaign Templates directly from the Campaign landing page

SugarCRM is a complete open source CRM (customer relationship management) solution used by many small and medium-size firms and organisations. SugarCRM is a lot more than just a CRM solution. It has great integration capabilities with other software and services, and can be used for several purposes. In this article we will use SugarCRM to set up, host and manage an email marketing campaign. You can send such a campaign directly from your mailbox, as people often do. The advantage that a solution such as SugarCRM gives you is that it adds things like unsubscription links, contact list management, monitoring of the campaign etc. As an end-user you get a lot more power with a little effort, making your campaign efforts more effective.

Advisor

Sukrit Dhandhania has spent several years working professionally, implementing several open source tools for companies. During this time he has evaluated, set up and maintained various open source tools for these firms

> **SugarCRM is a lot more than just a CRM solution. It integrates with other software and services**

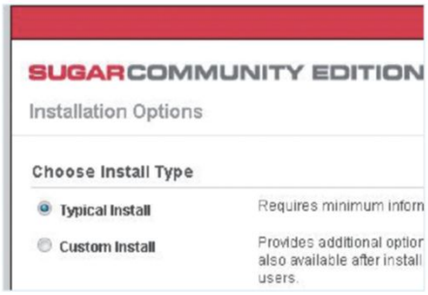
1. Get SugarCRM

SugarCRM is one the most popular open source customer relationship management solutions out there today. The software is pretty straightforward to download and setup. Head to the project's website, **www.sugarcrm.com**, and download the latest release of Sugar Community Edition. The latest release at the time of writing is 6.0.3 (**Fig 1**).

2. Set up SugarCRM

The prerequisites for setting up SugarCRM on your server are pretty straightforward. A server with a recent release of Linux, Apache, PHP and MySQL should suffice. Extract the file you downloaded in the previous step under your web server's document root. Access the directory through your web browser using the URL of your web server's document root. The installation

should begin automatically. You will need to create a database and feed in the database credentials during the setup.



■ The installation of SugarCRM is quite simple

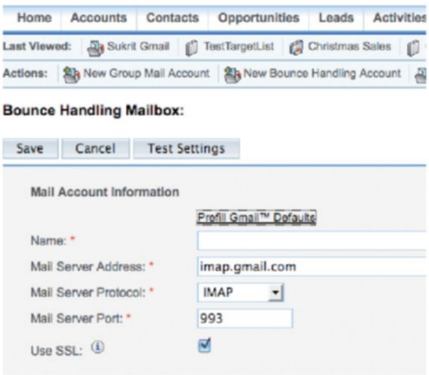
3. Post-installation

Once the installation is completed you should go about setting up SugarCRM. Create an admin user, enter your time zone, along with other details. Feel free to add other users and accounts, and add some contacts. The complete configuration of SugarCRM is beyond the scope of this article, so we will proceed assuming you have setup the CRM (**Fig 2**).

4. Set up email

As we are going to be using SugarCRM to send out and manage email campaigns, we need to make sure that the email functionality is up and running. There are different levels of configurations you can do for email. Let's look at one that is simple, and will suffice our requirements for this exercise. To quickly set up email, you need to click on the Admin link at the top of your page, then click the 'Inbound Email' link. Now click on the 'New Bounce Handling Account'.

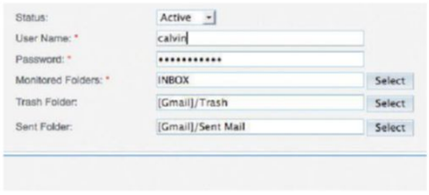
Fig 1 Get SugarCRM Get the open source release of SugarCRM



■ Add a new Bounce Handling Account

5. Set up email account

Enter the details of your email account. SugarCRM is pretty smart in this respect. It can, for example, import some email server settings from major email providers. In the screenshot below you will see a 'Prefill Gmail Defaults' option. Clicking on this option will fill in the server and port details for you if you are using a Gmail account to send out your campaigns. Enter the other details as per your email setup.



■ Set up an email account so SugarCRM can send out your campaigns

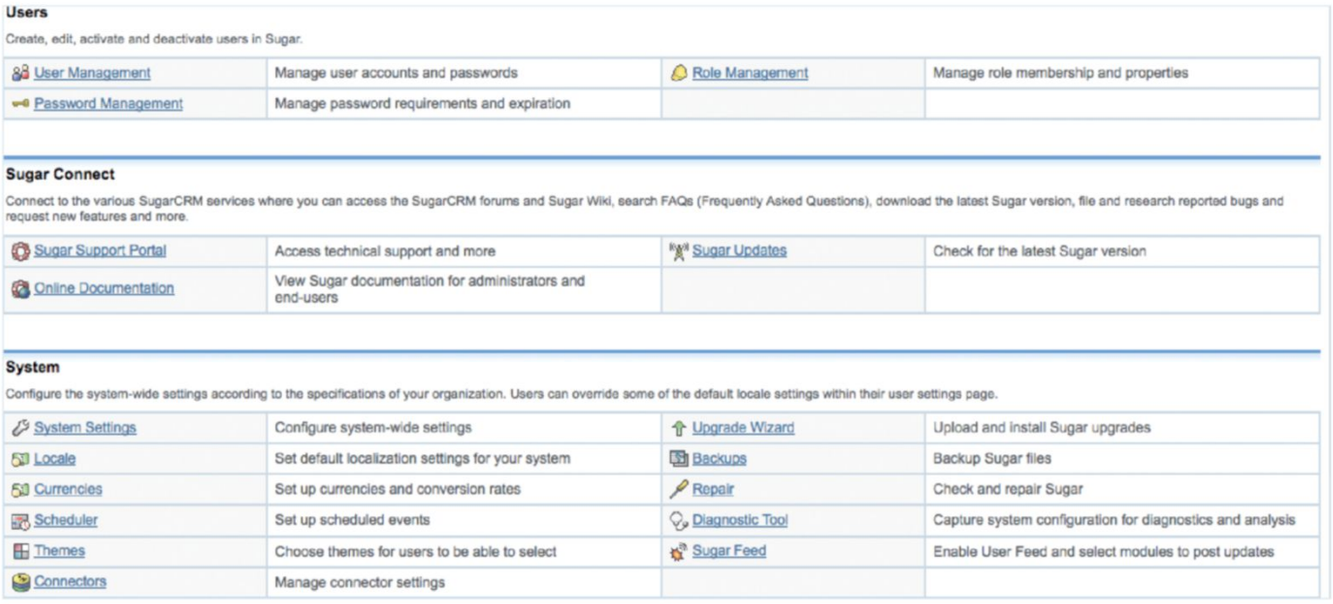
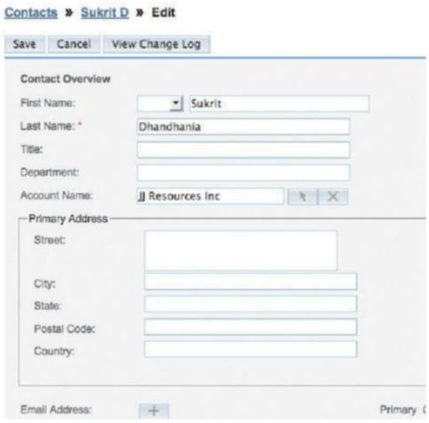


Fig 2 Post-installation The SugarCRM configuration and administration panel

6. Add contacts

Before we get cracking with setting up the email campaign, you need to make sure of one more thing – that you have some contacts set up. Click on the Contacts tab in the SugarCRM menu. Click on the 'Create Contact' button to create a contact. Enter the basic details such as the contact's first and last name. Make sure you also enter an email address for the user. Hit Save to finish this step.



Add a few contacts so that you have recipients for your email campaign

7. Launch the Campaign Wizard

Now that we have the basics set up, we can get on with setting up our first email campaign. Click on the Campaign link in SugarCRM's main menu. Now click on the 'Create Campaign (Wizard)' link from below the Campaign tab when the next page loads. This will launch the Campaign Wizard, which is the tool we will be using to set up our first campaign.

8. Different types of campaign

You will now be asked to select which type of campaign you want to set up. The three types are: newsletter, email and non-email-based campaign. Briefly, newsletters are recurring mailings that are sent on a regular schedule, Emails are for one-off style blasts, and non-email-based campaigns are just for tracking other non-email-based advertising you might be doing. We will use the second option for this exercise. So select the Email option and hit the Start button (**Fig 3**).

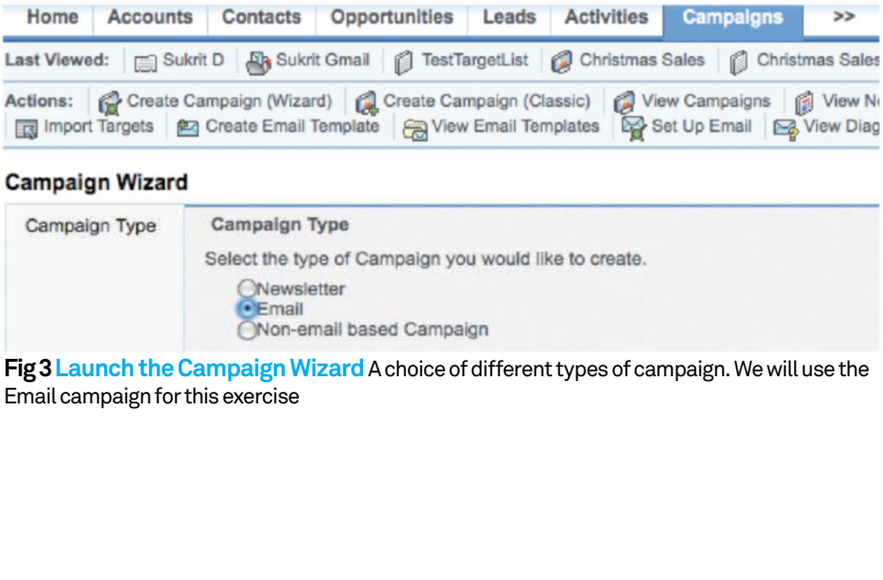


Fig 3 Launch the Campaign Wizard A choice of different types of campaign. We will use the Email campaign for this exercise

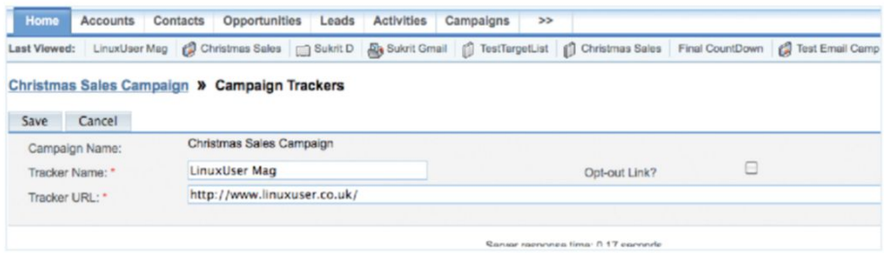
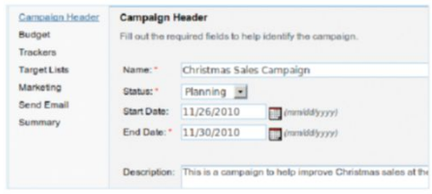


Fig 4 Campaign Tracker URLs Create a tracker URL and add a link to your website, or to the campaign's webpage, on it

9. Campaign Header

You will now enter the setup of the email campaign creation wizard. This involves a few steps. In the Campaign Header setup, at least enter a name for your campaign, along with the start and ends dates for it. The Status, Start Date and End Date fields are only required for records and will not affect the campaign. Hit Next when done.



Set up the Campaign Header

10. The budget

The next page is the Budget page. On this page you should enter your budget and revenue details. This will help you track the ROI (return on investment) for the campaign you are working on. However, if you are not using SugarCRM to track your budgeting, you can skip this step. Hit Next when done.

11. Campaign tracker URLs

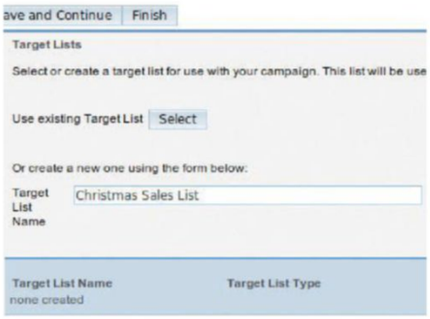
Your campaign emails can contain hyperlinks for things such an external website and even a link that users can click to be unsubscribed from the campaign. You can, for example, add 'LinuxUser Mag' as the tracker name and a link to this magazine's website by adding the link **http://www.linuxuser.co.uk/** for Tracker URL (**Fig 4**). Just make sure that you add the 'http://' at the beginning of the URL. When you are done, click on the 'Create Tracker' button.

12. Opt-out link

Aside from adding a link to an external website in your campaign, you can also add an opt-out link. This is a link that users can click to be unsubscribed. To do this, enter a suitable Tracker Name and check the 'Opt Out Link?' box (**Fig 5**). That's all you need to do. The Tracker URL for the unsubscription will be automatically generated for you. When you are done, click on the 'Create Tracker' button. Hit Next when you are done with this page.

13. Target lists

The 'target' while creating email campaigns is the people who will receive your campaign's emails. In this step you need to either use an existing recipient list, or create one. To create a new target list, enter the name and type of target list and then click Create. At this point you will see two options to proceed. There will be a 'Save and Continue' and 'Finish' button. You can exit out of the Wizard here using the 'Finish' button, but for the purpose of this exercise, click on 'Save and Continue'.



Add a new target list

14. Marketing Email

On the Marketing Email page of the campaign setup, we will set up the details of the emails that we will be sending out. Enter a name, pick an email account that should be used to send out all the mails, select a list, and add a date and time for the campaign to go out (**Fig 6**). The Email Template bit is a bit more complicated. We'll look at it in the next step.

15. Email Template

When you click on the drop-down menu for the Email Template, you should see a few options. You can, however, set up something of your own. Click on Create. You will see a number of fields along with a fancy text editor. This is where the magic of SugarCRM's email campaign comes in. You can use variables such as $contact_name (click on the Insert button next to the $contact_name option) and these variables will be replaced with the real values when the emails are being sent out. These templates can be reused over and over again in different campaigns, so spend some time working on this (**Fig 7**).

16. More customisation

Once you have entered some text along with links and variables, it's time to make things a bit nicer. If you look at the screenshot from the previous step, you will notice that the text on hyperlinks doesn't really work out too well. Let's change that. Click on the HTML button at the top left of the rich text editor. Here you can make changes directly to the HTML code. Hit 'Update and Save' when done.

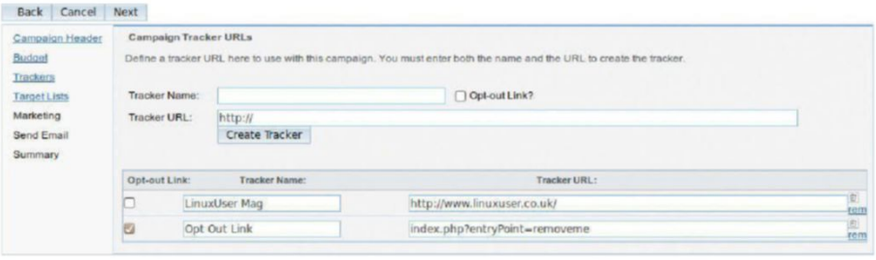


Fig 5 Opt-out link The campaign tracker URLs all set up

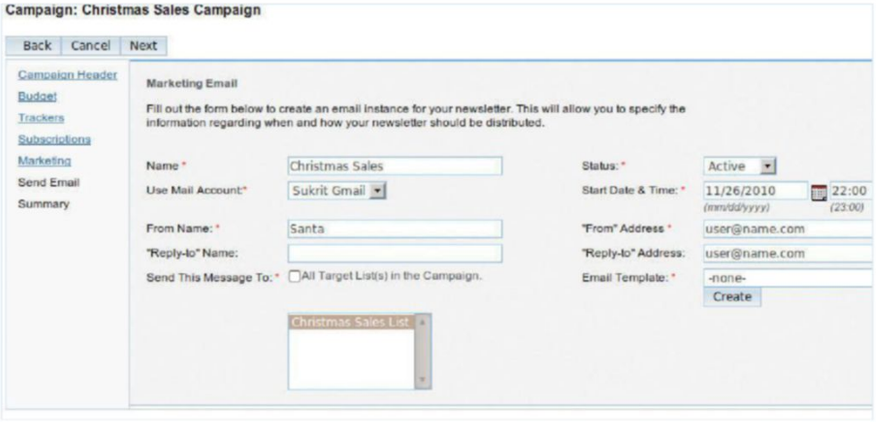


Fig 6 Marketing Email Configure your marketing campaign mail with the outgoing mail server and the sender's email ID

17. Send test email

Now you should return to the Campaign Creation Wizard page. You will now have the option of sending out the email as a test. Do it! Check and confirm that you receive the email and click to test both the LinuxUser and opt-out links in the email you receive. If you are using a web-based email client, make sure you allow the images to download so that the act of opening the email will be tracked by the tracker. Once the test works, you can send out the real campaign mails.

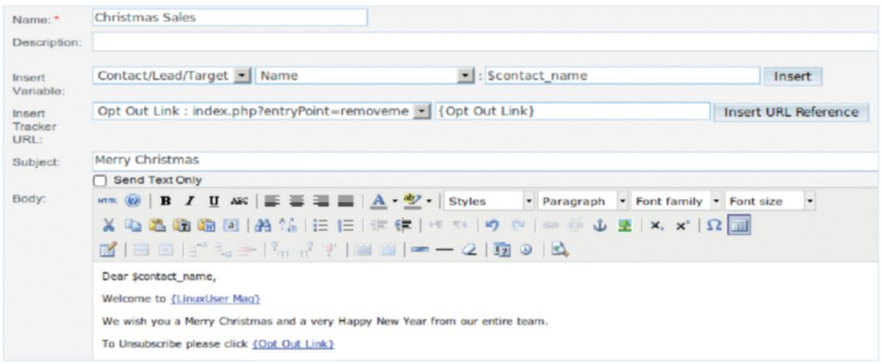


Fig 7 Email Template This is where you need to spend some time and create an attractive email template

18. Working around email limits

When you are using a public email server to send your campaign emails, you might find that the server has a cap on the number of emails you can send out at the same time. To work around this possible problem, SugarCRM has the ability to break up the total number of emails and send them out at intervals. To access this configuration setting, go to Admin>Schedulers>'Run Nightly Process Bounced Campaign Emails' in SugarCRM. Here you can configure SugarCRM to send out emails every hour, on the hour. You can also set a limit on the number of emails SugarCRM should send out in every burst by going to Admin>'Manage campaign email settings'. The value here is set to 500 by default.

Linux myths: time to put the record straight?

There are still many false roadblocks put in the way of many potential Linux users. Jon Foster looks to set the record straight

To outsiders looking in, there's still a conundrum of uncertainty surrounding Linux. There are constant complaints, fears and thoughts raised, that all seem to raise a virtual firewall stopping people from at the very least giving it a try.

But what are these myths? What are people saying about Linux, and is there any substance to the complaints being raised about it? Bluntly, what happens when a newbie heads in your direction spouting one of the following?

Let's take a look to see if there is substance behind the myths...

MYTH 1:

Linux is hard to install

The very idea that Linux is a tricky operating system to get up and running is simply laughable...

The simple truth here is that it's hard to imagine an operating system anywhere that can be as straightforward to install as many variants of Linux. Take the installer for Ubuntu as an example. That's much, much easier than Windows to get up and running, and it has the courtesy to ask you proper questions, in an easy-to-understand manner, to help you get from A to B.

Furthermore, once it's installed, more often than not there's nothing else you need to do. Drivers are in place, software is ready to go, and there's no hunting around for patches and anti-virus software as your first job. A win, surely.

MYTH 2:

Linux is very hard to use

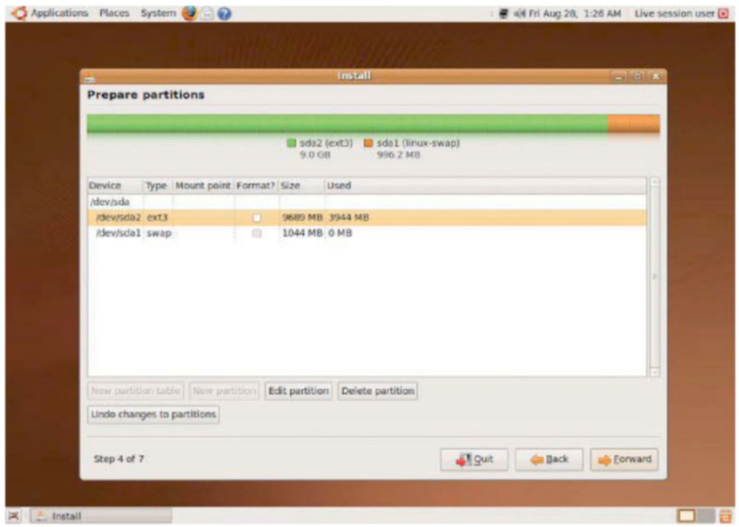
Well, if you can't swat this one aside, we're already in trouble...

The ongoing perception that Linux is difficult and tricky, and will leave most users perplexed, is one that the mainstream press has been perpetrating for many years. Just look at the stories that followed the original netbooks arriving in PC World with Linux installed. Then, we got tales coming out that return rates in such machines were higher, because people wanted Windows. Linux was bad, Windows XP was here to save the world. Just marvellous.

So where do we start? Well, Linux can be hard to use. There's no point denying it. If you go for an advanced distribution, or want to delve around the very guts of the kernel, then it can be a tricky beast. No more so than any other operating system in that respect, but nonetheless, it's worth accepting there's a sliver of truth to the argument.

But just a sliver. Because as anyone who's installed Ubuntu, or a distro of that ilk, can testify, Linux is a breeze. The installation is faster, more streamlined and easier than a Windows installation. You get to a working desktop in double quick time, replete with most of the applications you need to get working (and the others are but a click or two away). The old days of faffing around with networks, sound card drivers and display adapters are gone. And Linux works, quickly, right out of the proverbial box. Choose the right distro, and it's really not hard to do so, and the ease of use easily matches that of Windows.

What the core of the 'Linux is hard' argument usually boils down to is 'Linux is not Windows' (someone who has never used Windows before is likely to find it tricky, by the same thought process). But then, isn't that part of the point?



Ubuntu installation is as easy as pie

MYTH 3:

Open source software is free, and thus not very good

All software runs the risk of not being much cop. But few people ever sit down to write an application with the hope that it won't ever get used

It is fair to say that there's a lot of free open source software to wade through, and not all of it is worthy of your time, or up to the standard of a commercial application. Yet there's a meritocracy at work here, and it's not tricky to find out which software is worth trying. Many distros include such recommended titles in their repositories, too. And the beauty of a quality piece of software for Linux is that it never comes with the massive price tag, the constant nagging to register, or the push to make you 'upgrade' one year later.

> "As anyone who's installed Ubuntu, or similar, can testify, Linux is a breeze"

MYTH 4:

I don't want to use Linux, because I don't want to use the command line

So don't use the command line - you don't have to!

This is the one that really gets under the skin, for a couple of reasons. Firstly, and perhaps most importantly, just because you use Linux doesn't mean, by definition, that you have to open up a terminal and start inputting large swathes of code. It's patently nonsense to think that. There are many Linux users who've never got anywhere near to the command line, who use the OS with a graphical user interface with the minimum of fuss. It's easy to do so. If anything, treat the terminal as a shortcut to getting things done, but go through the GUI if you prefer - the choice, as they say, is all yours.

Secondly, though, what's actually to fear about the command-line interface? It's certainly a lot friendlier than tapping CMD into Windows. You can copy and paste code from other windows for starters, which is a massive help that many users seem not to realise. Plus, working with a terminal is nowhere near as difficult as fearmongers may have you believe. Okay, granted, there are some for whom typing in commands is, and will always be, alien. That's fine. But Linux, as it always does, offers the choice. Which leads us on to...

World Of Padman is just one of the many games available on Linux

MYTH 5:

There's too much choice

Easily one of the most bizarre arguments on planet Earth, and yet there it is...

There are some who criticise Linux for precisely what it brings to the world: options. You can see the thinking, to be fair. Some, when they get hold of a computer, just want it all set up, and all the key decisions made for them. That's entirely fine, and understandable. But the argument that people are thinking of trying Linux, but don't know where to start, needs to be addressed.

Is there a modicum of truth to it? Yes, of course there is. Dell runs big adverts where it states at the top that it recommends Microsoft Windows. Some people like that. And ask them to pick a Linux distro from the hundreds available and panic sets in. So perhaps the Linux community needs to rally around an entry-level distro, as it sort-of-has with Ubuntu, and push that as a starting point.

As for the notion of offering choice? Well, you're damned if you do, damned if you don't.

MYTH 6:

Linux isn't compatible with other computers

It is. Perspective is crucial here...

Linux is an operating system. It's software that sits atop a collection of components. It does have things in common with other operating systems.

Thus, somewhere along the line, someone did actually think about cross-compatibility. Quite a lot, as it happens, as it's genuinely hard to come up with a file format that Linux doesn't support in one way or another. You get the odd, very extreme proprietary format that's hard to bash into an open source program, and it's not the open source community that's to blame for that. It's still the reality of the situation, though. Yet there are still very few file formats that don't enjoy some degree of support, and Linux cross-compatibility is arguably better than any other platform.

While we're here, it's worth having a look at the compatibility simply between different versions of Windows. There are applications that work in Windows 2000 that won't work in later editions, and there's a decade's worth of progress in there. Is that something that's deemed acceptable? Sadly, seemingly it is. Yet when people talk compatibility, Windows – since it's many people's default – seems to be exempt from criticism.

MYTH 7:

The world uses Windows, and so should I

And that, friends, is why nothing changes...

It's why status quos stay as they are, and it's why the number of complaints about using Microsoft software, as if there's no tangible alternative, continue to mount up.

It's rubbish, too. Many major companies rely on Linux servers for their business-critical applications, not least for the sheer security factor. Furthermore, what the world actually uses is a series of file formats which can easily be read across different platforms for the most part, as we've already discussed. Going along with something simply because it's the most popular is a dangerous strategy, and one that's costing many people unnecessary funds.

The world doesn't have to use Windows. In the mobile market, Windows is heavily in the minority. The status quo, at last, might be changing, and a move away from proprietary software may yet be in the offing. But not while people are peddling guff that it's a Windows world as a sole reason for jumping aboard the bandwagon...

MYTH 8:

You can't play back Blu-ray discs on Linux

No. That's true. You can't...

You can't do that because Blu-ray is a proprietary format, and if anyone were to try to break it so that you could legally buy Blu-ray discs and play them back on a Linux machine, then you could get prosecuted. Instead, if you want to play a high-definition film off a Blu-ray disc on a Linux operating system, it has to be a pirate copy. Which is illegal. And we don't condone that, of course.

It's a nonsense, but if you are desperate to get Blu-ray up and running on Linux, legally, then you've got a potentially never-ending wait.

MYTH 9:

I can't get printers to work with Linux

This was a bugbear for a while, to be fair, as while many people had few problems with printers and Linux, a good chunk nonetheless were struggling with compatibility

Yet it's another example of how a fast, reactive, open source community can resolve a problem quicker than any multinational conglomerate could. Look at the work that's gone into sites such as **openprinting.org**, and you begin to appreciate just how hard the community works to resolve problems.

We've cited printer problems here, but realistically, we could have picked drivers to most products. Because a user's biggest problem with a computer is generally the one that's in front of them at the time, and when it's drivers, few issues tend to be more frustrating. Linux? It's got answers, and lots of them, and helpfully, you're not waiting on a manufacturer or a big company to wave their magic wand and solve whatever issue you have.

MYTH 10:

Linux has got too bloated

Actually, to be honest, there might be something in this

Linus Torvalds himself has admitted in the past that Linux is bulking up a little, as the size of the core kernel has grown. And given that the market is moving towards mobile operating systems, there are some questions worth asking there.

Yet surely, everything is relative. Microsoft may have downsized the bloat of its Windows operating system with the move from Windows Vista to Windows 7, but it's still a massive installation by anyone's measure. Linux is much more streamlined at heart, and while you can find distros that do give you a DVD chock full of features and software you might not need, that's the exception rather than the rule.

Still, expect some kernel pruning to be on the agenda in the future, as Linux continues to retain its competitive edge in this department.

Your turn

What Linux and open source myths get your goat? Write in, and we'll print a selection of them in our letters page in the issues ahead...

linuxuser@imagine-publishing.co.uk

MYTH 11:

There's no technical support if something goes wrong!

What it usually amounts to is that there's no call centre on the other side of the planet to play on-hold music for hours on end...

That there's no elongated phone call with a technical support advisor. That there's no batphone to reach for if things go wrong. It's the perception that there's some kind of comfort blanket missing, and it's an argument cited even by those who never use technical support.

Firstly, it should be noted that some Linux distributions do have formal technical support. Let's get that out of the way first. More importantly, though, what there is instead is the largest community of support for any operating system on the planet.

Each of the hundreds of variants of Linux on the planet has a support network online that'd rival any product on the planet. What's more, it's mainly powered by goodwill. The end result of this is that if you have a Linux problem, just searching online is likely to provide you with an answer, and quickly. Granted, some explanations are choppier than others, but that's something that can be levelled at any level of technical support. However, there's an old adage with Linux that every question you have to ask about it has already been asked by someone, somewhere. And by extension, it's already likely to have been answered. Even if you can't locate what you're hunting for, just ask. You won't even have to dial a premium-rate phone number to do so.

MYTH 12:

I can't run my Windows applications on Linux

Yes you can. In fact, the lion's share of Windows software is possible to get up and running on Linux quite easily

All you need, as it happens, is a friendly, monthly Linux magazine, that just happens to have a tutorial on the very matter in the same issue as this feature. That'd be a master class in planning right there.

It's a myth and a half, to be clear: most things you wonder whether they're possible with Linux *are* possible with Linux. It's often that simple.

MYTH 13:

Linux is rubbish for games

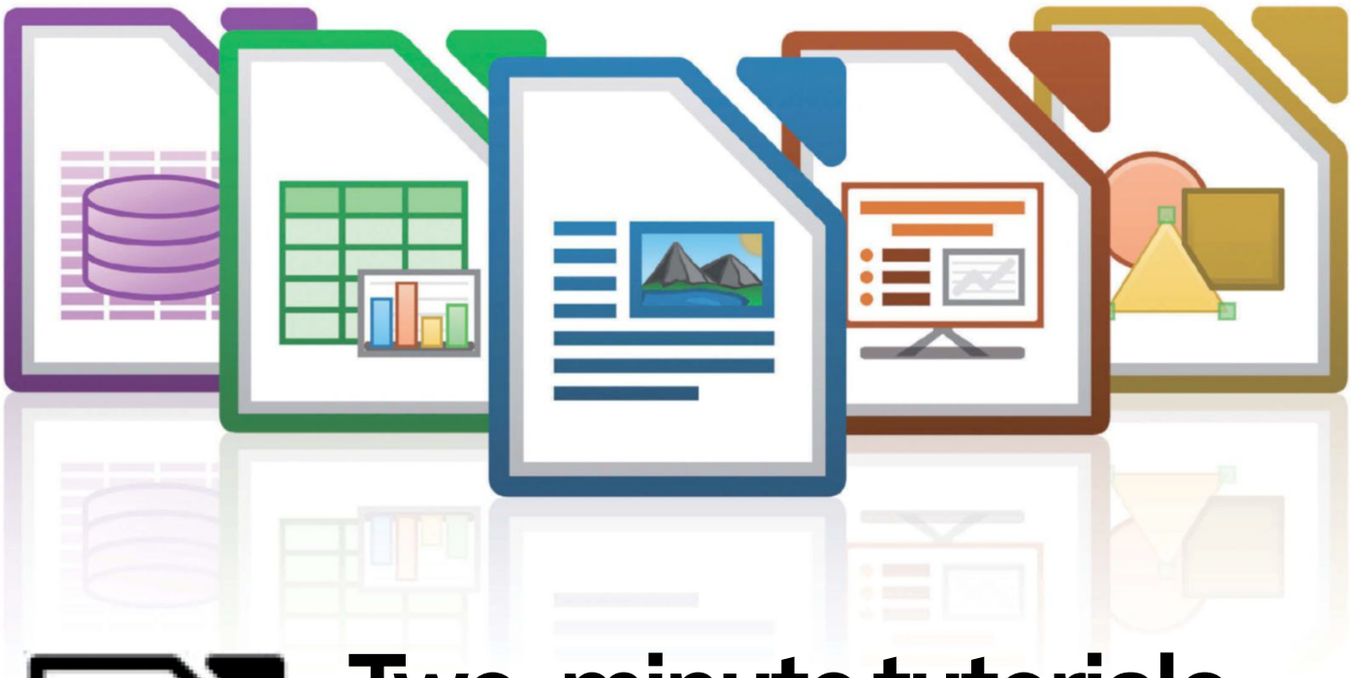
No it's not. Linux is perfectly fine for games, but it depends what games you want to play...

Many Windows games will work, to a degree, through an emulator or translation layer such as Wine. And then there's a vibrant, busy gaming development scene specifically for the Linux OS.

The gaming problem generally surrounds the fact that the very latest, cutting-edge games don't run on Linux, and that's true. There's no point trying to run *Need For Speed: Hot Pursuit* on Linux, Wine or no Wine. Yet Windows gaming isn't at its peak, either, and as game publishers look to exercise more and more control, a greater number of them are heading for consoles.

Linux, meanwhile, has a sizeable catalogue of games, catering for all genres, and with no price tag attached. Not for the first time, the perception here is off the mark, providing you are willing to readjust exactly what it is you're expecting.

Want proof? Then you should take a look at: **www.playdeb.net/updates/ubuntu/10.10/ http://en.wikipedia.org/wiki/Linux_gaming www.penguspy.com** or **http://www.linuxlinks.com/article/20080510052539217/Games.html.**



Two-minute tutorials – LibreOffice

A selection of quick tutorials for basic and advanced techniques in the popular LibreOffice suite

Advisor

Ken Hess is a Linux and open source author and columnist who writes on a variety of topics from applications to databases to in-depth how-tos... www.kenhess.com

Welcome to the new two-minute tutorials series. This series focuses on how to connect with and enjoy Linux applications and features. The features explored range from beginner to advanced, so there's something to discover for everyone at every level. Regardless of the experience level, the tips can be explored and completed in two minutes. The mission is to assist end-users who are transitioning from Windows, to introduce new features, to enhance knowledge, and to enjoy the Linux experience more fully.

Two-minute tutorials are presented in stepwise tutorial fashion so that anyone can easily follow along and learn something new. The series covers topics such as office applications, shell scripting, system administration, peripherals, networking, services, connectivity and much more.

Working with LibreOffice Writer and graphics files

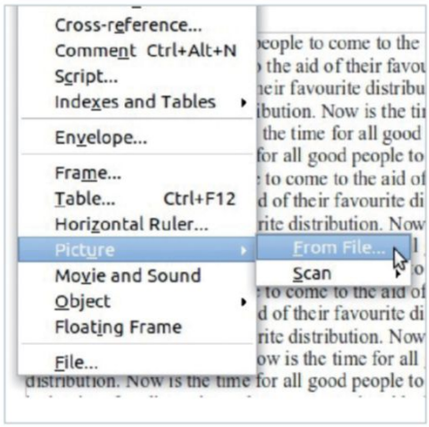
Learn how to insert and manipulate images effectively in LibreOffice Writer

Mixing image files and text in word-processing documents can be absolutely maddening, but this two-minute how-to makes it easy.

1. Insert the graphic

Select Insert>Picture>From File from the Writer menu. Browse to the image file, select it and

click Open to place the file into the document.



■ Placing a graphic image into a document

2. Adjust the size of the image

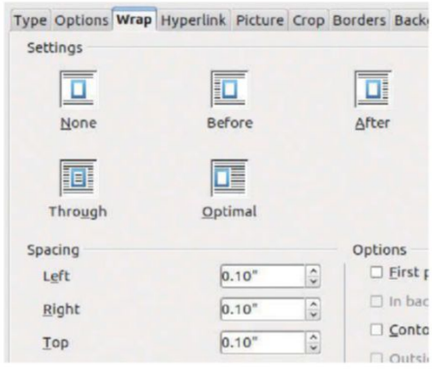
Right-click the image that is now in the document and select Picture from the list. Select the 'Keep ratio' option and adjust the width or height of the image. The image adjusts accordingly. Note that resizing with the mouse does not maintain the aspect ratio (the correct image proportions).

3. Moving an image within a document

Grab the image with a mouse left click and move the image to the desired location within the document.

4. Change the image and text relationship

Right-click the image, select Picture and then select the Wrap tab. The default option is Optimal. Choose other options to create the desired image-to-text relationship.



■ Changing how text interacts with the embedded image

5. Add a caption to the image

Right-click the image and select Caption from the list. Enter the image caption in the Caption field, select a category from the list, and click OK when finished.

Create a PDF version of a document

LibreOffice has the built-in capability of creating PDF files from documents

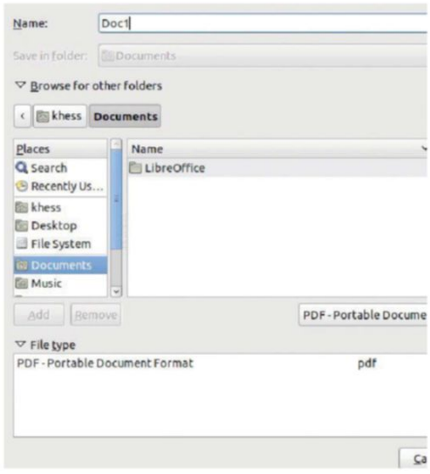
Those who have a need to create portable document format (PDF) files from standard text documents will appreciate the built-in capability to do so in LibreOffice.

1. Open the original document

Open the document that requires conversion to the PDF format in LibreOffice and left-click the 'Export Directly as PDF' button on the toolbar.

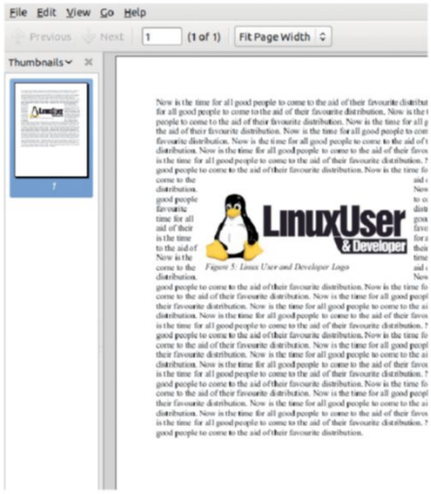
2. Name and save the PDF document

Enter the name of the converted document, select 'PDF – Portable Document Format (.pdf)' and click the Save button.



3. Examine the PDF in Xpdf

Open the PDF file in Xpdf, or another PDF reader, and examine the new file for correct formatting and non-text entity placement and size. Alter the original document and repeat the export, if necessary, to achieve correct formatting in the PDF.



Non-printing characters in documents

Learn how to avoid frustration when working with non-printing characters

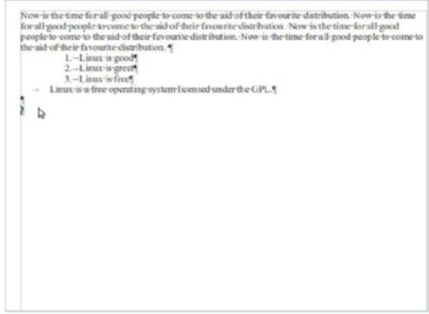
Editing documents with non-printing characters can be frustrating. Learn to see and edit the documents in their true form.

1. Reveal a document's non-printing characters

Locate the non-printing character symbol (¶) on the menu and select it. To hide the non-printing characters, select the symbol again.

2. Edit the non-printing characters

Now that the non-printing characters are revealed, it's easy to edit a document without the odd shifting about that sometimes occurs during the process.



Using drop caps in documents

A drop cap is more than simply an enlarged character; it's a special paragraph format

When used sparingly, once per document or once per chapter, drop caps can enhance the look of a written work.

1. Enlarging the first character

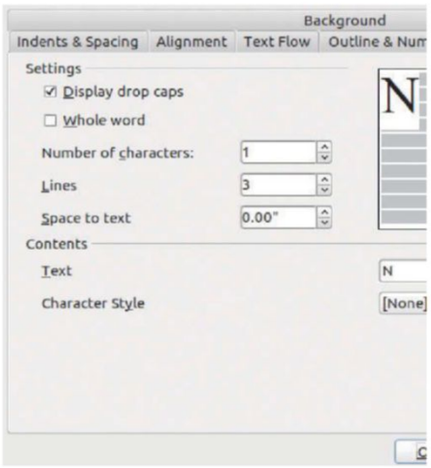
Open a document, select the first character and increase the font size to 36 point. This example is not a drop cap. Undo the enlargement.

2. Formatting the paragraph

Select Format>Paragraph to open the Paragraph attributes window.

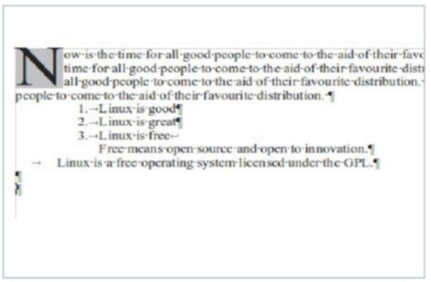
3. Creating a drop cap

Select the Drop Caps tab, check 'Display drop caps' box. The default settings are a good start.



4. Checking the drop cap insertion

Click OK, when finished configuring the drop cap parameters. You can adjust parameters by repeating steps 2 and 3, if necessary.



Separating text into columns

Columns can break up documents that contain mostly text, to make for easier reading

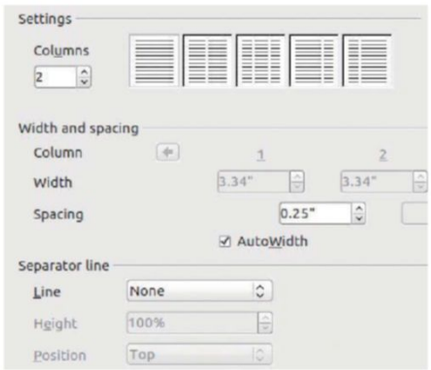
Documents that are text only or mostly text tend to make readers skip around, potentially missing important information. Break up text with columns.

1. Configuring column parameters

Open a text-filled document and select Format>Columns from the menu to open the Columns properties window.

2. Setting column properties

Change the number of columns to two and change the spacing (between columns) to 0.25". Note that the preview changes as parameters are altered.



3. Checking the column changes

Click OK to accept the changes. The entire document converts to a two-column layout. Return to the Columns properties window to make changes.

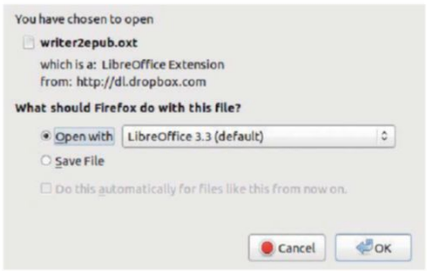
Working with extensions in LibreOffice

Extensions add increased functionality to LibreOffice applications at no cost

LibreOffice and all of its extensions are free (non-proprietary) software. This how-to demonstrates extensions by installing the Writer2ePub extension. ePub is the Kindle document format.

1. Download the Writer2ePub extension

In your web browser, go to **http://libreplanet.org/wiki/Group:OpenOfficeExtensions/List**. Select the Writer2ePub extension and click 'Get it!' Click OK at the prompt to 'Open with LibreOffice 3.3'.



2. Install the extension

LibreOffice warns that an extension is about to be installed. Click OK to accept and proceed with the installation.

3. Accept the EULA

Read and accept the end-user licence agreement by clicking the Accept button to continue.

4. Completing the extension installation

Note that the extension is correctly installed in LibreOffice. Click Close when finished.

5. Locate the extension

The Writer2ePub has added its own toolbar to LibreOffice with three buttons. Extensions generally integrate into LibreOffice as toolbars or as menu entries.

Autofill data series for Calc

Creating an autofill data series saves time by saving keystrokes

This how-to demonstrates how to insert a series of numbers into a spreadsheet by supplying a single number and a pattern for Autofill.

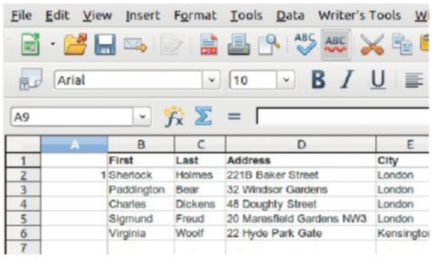
1. Preparing the list

Open a contact list or other list of entries in a spreadsheet that need numbering. Numbering of each row of data is important for database conversion. Select column A to highlight it.

2. Insert a new column

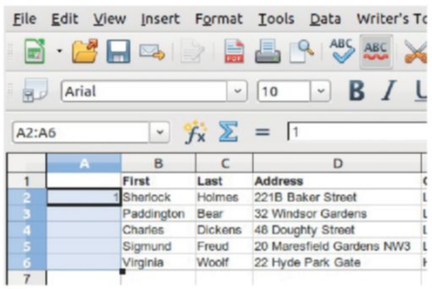
On the menu, select Insert>Columns. This action

creates a new column A and moves everything one column to the right. Type the number '1' into cell A2.



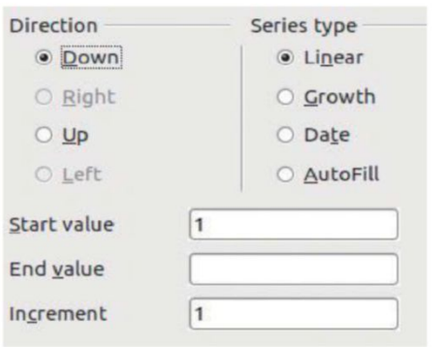
3. Select the row range

Left-click cell A2 and drag down the list until the entire range of rows is highlighted. This instructs Calc how far to process the series.



4. Define the series

Select Edit>Fill>Series from the menu. For this demonstration, select Direction: Down, Series Type: Linear, Start Value: 1, and Increment: 1. This will number each row with normal counting numbers.



5. Complete the series

Click the OK button, when you have finished editing the series parameters. The numbered series inserts into the column.

Quick sums and averages

Learn to use two handy maths functions made easy with Calc

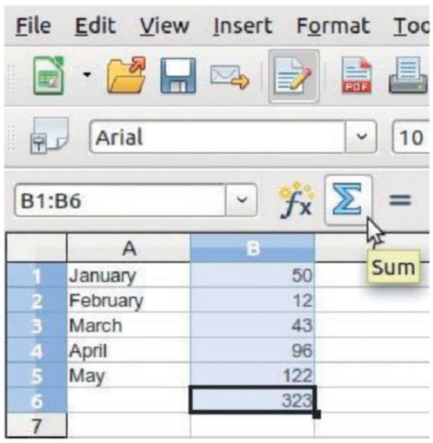
With a few mouse clicks, you can add impressive maths functions to spreadsheets for a professional touch.

1. Select column of numbers

Open a spreadsheet containing a list of numbers requiring a sum. Highlight the list with a mouse left click and drag.

2. Creating the sum

Click the Sum (Σ) button to add up the column of numbers. The sum will appear directly below the highlighted column of numbers.



3. Inserting an average

To add an average of numbers to a spreadsheet, select the cell where the average is to be located and click the Function Wizard (fx) button.

4. Using the Average function

Scroll down the function list, find AVERAGE, select AVERAGE and click Next to continue.

5. Enter the cell range

Into the formula '=AVERAGE()', enter the cell range B1:B5, then click OK to complete the function wizard.

6. Enter labels for sum and average

Add text labels to the spreadsheet for the sum and average, for easy reference.

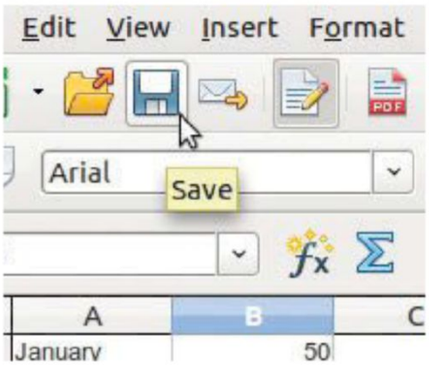
Working with Microsoft Office users

LibreOffice has built-in Microsoft Office compatibility for those who haven't tasted freedom

To help with transitioning from Microsoft Office, and to remain compatible with non-LibreOffice users, compatibility with other formats is included.

1. Save the original file

Save the LibreOffice file in its native format (ODT, ODS, ODP) by selecting the Disk (Save) icon on the toolbar. Name the file and click Save.

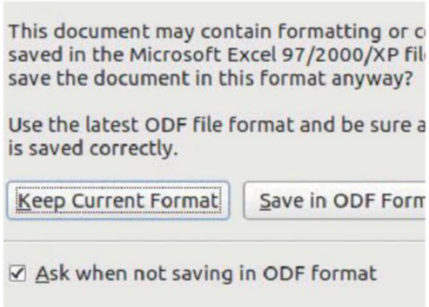


2. Save As Microsoft Office document

Save in a Microsoft-compatible format. Select File>Save As. Name the file and Save. LibreOffice will append the correct filename extension (DOC, XLS, PPT).

3. Acknowledge format warning

When saving in a foreign file format, LibreOffice warns that some document formatting may be lost due to basic incompatibility. Select Keep Current Format to continue.



Two-minute tutorials – virtualisation

Advisor

Ken Hess is a Linux and open source author and columnist who writes on a variety of topics, ranging from applications to databases to in-depth how-tos...
www.kenhess.com

Create better virtual machines by learning these performance-boosting secrets...

Welcome to the second instalment of Linux User & Developer's two-minute tutorials series. This month's focus is virtualisation. Virtualisation makes it possible to maximise hardware resources by running more than one operating system per physical system. It is the basis for cloud computing and is a mature and a ubiquitous data-centre technology. Virtualisation, especially at the enterprise level, includes Windows operating systems, but don't be put off by this; it is a simple fact of managing a large environment.

This month's tutorials demonstrate various methods of optimising a virtual machine (VM) environment. Optimisation techniques cover VM memory, CPU, disk and network. Other how-tos cover working with snapshots, and VM deployment from templates.

These tutorials include virtualisation technologies from VMware, Oracle and Citrix. The techniques are not restricted to one virtualisation vendor, software or operating system. The terminologies may differ a bit between vendors, but the concepts are vendor agnostic.

All techniques herein are current, practical and practised in data centres and on desktops worldwide. These virtualisation how-tos were written using the latest available software versions: VMware Workstation 7.1, VirtualBox 4.0.4 and XenServer 5.6.

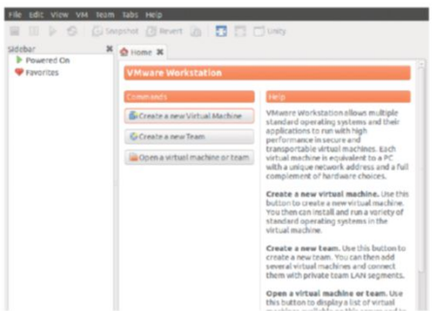
VMware Workstation new virtual machine construction

Creating new virtual machines is easy, but there are optimal settings when doing so

Anyone can create a virtual machine using a wizard to step through the process, and the default settings are good but not great. Learn how to create optimally performing virtual machines in this how-to.

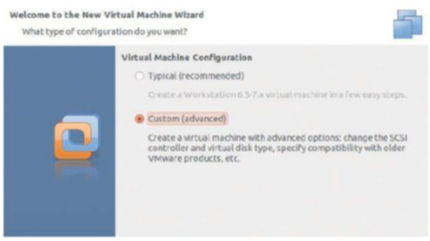
1. ESX compatibility

Create VMware Workstation VMs with ESX compatibility in mind. Doing so creates VMs with the greatest scalability, compatibility and flexibility. To begin with, select 'Create a new Virtual Machine'.



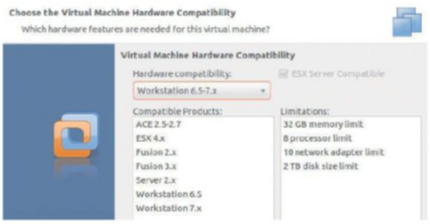
2. Customise virtual machine configuration

When creating a new VM, always use the Custom (advanced) selection to maintain the greatest amount of control over configurable settings. Select Custom and Next to continue.



3. Selecting virtual machine compatibility

Select Workstation 6.5-7.x for the widest possible range of compatibility and scalability for new virtual machines. ESX Server compatibility is automatic when using this setting.



VM CPU configuration

Learn how to select an optimal CPU configuration for virtual machines

Begin with a single CPU and single core per processor. Increase this only after performance observations. The VM host won't allow the maximum number of cores or CPUs to be exceeded; since it is a dual-core, single-processor system, no VM may exceed this value.

1. VM processor selection

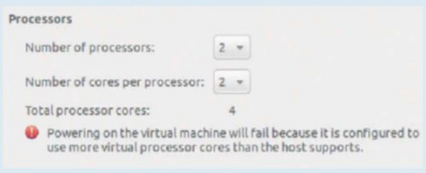
Select one processor (CPU) and one core per processor. Note the total processor cores value.

2. Increasing the number of cores per processor

Increase the number of cores to two for a single processor, or the number of single-core processors to two for two processor cores.

3. Exceed number of host cores

This step demonstrates what happens when the number of host processor cores is exceeded for a virtual machine.



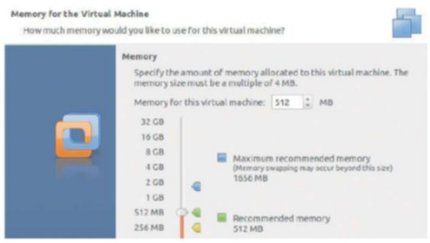
Memory, network, controller and disk configuration

Learn how to make optimal selections for non-CPU virtual hardware components

Virtual hardware selections can affect performance. This tutorial demonstrates wise selections for optimal virtual machine workload performance.

1. Virtual machine memory allocation

Based on operating system selections, VMware Workstation preselects minimum, recommended and maximum memory allocations. Use the recommended value.



2. Selecting the network connection type

A bridged connection provides a true network connection to the virtual machine and mimics a physical machine. A NAT type provides a high level of security. Host-only is a private network between virtual machines. Select the 'Use bridged networking' optiojn.

3. Choosing a SCSI disk controller type

Use the recommended SCSI disk (I/O) controller type for the virtual machine. Then click Next to continue.

4. Select a virtual disk type

Select the recommended (SCSI) virtual disk type for the virtual machine for best performance. The SCSI disk selection also matches the SCSI disk controller selected in the previous step.

5. Virtual disk configuration

For best performance, select 'Allocate all disk space now' and 'Split virtual disk into multiple files'. Allocating all disk space now will use the most disk space initially, but will result in better performance for the VM. This is also known as 'thick' provisioning.

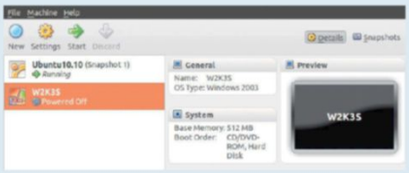
Adjust Oracle VirtualBox virtual machine display settings

Learn to adjust virtual machine display memory for best performance

Virtual machine display settings significantly affect the user experience. Video memory is drawn directly from the host's available memory pool. Increasing video memory has a significant effect on a virtual machine's interactive performance but generally, 32MB should be enough for any VM.

1. Select display settings

With the VM powered off, select Display from the Details view. The VM must be powered off to adjust video memory settings.



2. Increase video memory

Using the slider or the memory value field, increase the video memory to 32MB. Optionally, select 'Enable 3D or 2D video acceleration'. Click OK when finished.

3. Power on the virtual machine

Note the change in video memory for the VM. Power on the VM.

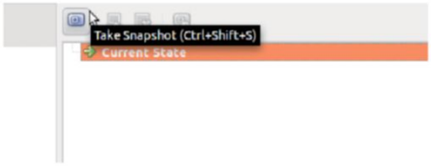
Taking VM snapshots

Make a quick virtual machine backup by taking a snapshot

A snapshot is a VM backup that can be performed on a live VM. It's a quick method of preserving state prior to installing software, applying patches, or adding drivers. Revert to a particular snapshot when a restore point is needed. All virtualisation technologies include the ability to perform snapshots.

1. Take the snapshot

On the main VirtualBox screen, select Snapshots and then click the Camera (Take Snapshot) button on the right pane. Note that the VM is running and shows Current State to distinguish it from snapshots.



2. Name and describe the snapshot

A window appears, prompting a name and description of the new snapshot. Enter a name and description for the snapshot, based on its current state.

3. Complete the snapshot

Click OK to save the snapshot and observe that the new snapshot exists in the snapshot list for the virtual machine.

4. Restore a snapshot

To restore a snapshot to the current state, select the snapshot from the snapshot list, right-click the snapshot and select Restore Snapshot.

5. Confirm snapshot restore

Confirm the snapshot restore point by clicking Restore on the warning screen. Please note that the current machine state will be lost and cannot be recovered.

Installing Linux from a URL in XenServer 5.x

No ISO required for a full install

XenServer allows installation of a VM via a URL, ISO or CD/DVD image. In this demo, we use the URL option.

1. Create a new virtual machine

Select New VM from XenCenter's home screen to launch the New VM wizard.

2. Select an operating system

Scroll through the list to find an operating system to install into the new VM. Select the operating system and click Next to continue.

3. Select installation media

Name the virtual machine, click Next to continue, then enter a URL from which to install the operating system into the new VM. Find installation URLs on the selected distribution's website. Choose a mirror site from the provided list of URLs. Experimentation may be required to find the correct URL depth for installation.

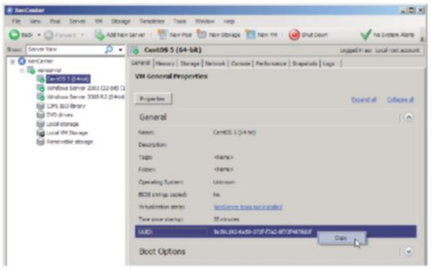
Correcting an errant installation media URL

How to correct a mistyped URL during VM configuration

If installation fails due to an incorrect URL, the VM must be removed and re-created; however, it is possible to salvage the VM.

1. Locate and copy the VM's UUID

On the XenCenter home screen, select the VM's General tab. Locate the UUID in the right pane. Select and copy the UUID with a right click of the mouse.



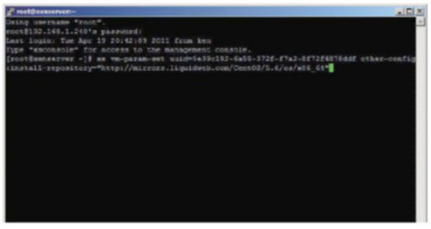
2. Connect to the XenServer

SSH and login to the XenServer's interactive shell. Login as the root user using the root user's password. If SSH access is not configured for the XenServer, the next tutorial demonstrates its setup.

3. Change the installation URL

Issue the following command to change the installation URL:

xe vm-param-set uuid=5e39c192-6a59-372f-f7a2-8f72f4878ddf other-config:install-repository="http://mirrors.liquidweb.com/CentOS/5.6/os/x86_64"



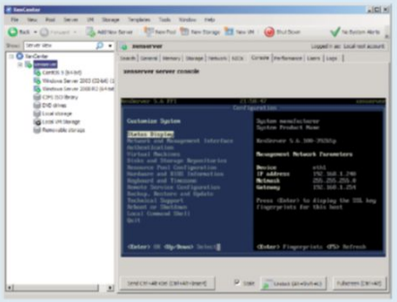
XenServer SSH access setup

How to set up remote access via SSH to the XenServer

Full administration of a XenServer virtualisation environment requires shell access to the XenServer host systems. Extreme caution must be taken when interacting with the XenServer host in this manner, since the root user is the all-powerful administrative account.

1. Connect to the XenServer console

Physically interact with the XenServer at the console, or use the console via XenCenter. Using XenCenter, select the XenServer in the left pane and then select Console in the right pane.



2. Select Remote Service Configuration

Using the keyboard arrow keys, arrow down to the Remote Service Configuration selection and press Enter to begin remote access configuration.

3. Select Enable/Disable Remote Shell

Arrow down to select Enable/Disable Remote Shell and press Enter to continue.

4. Enable remote access

Press Enter to enable remote access on the XenServer.

5. Confirm remote shell (SSH) configuration

Press Enter to accept the successful configuration of the XenServer remote shell. Press Esc to return to the main XenServer console screen. The XenServer system will now accept remote SSH logins.

Creating a XenServer ISO repository

Set up an ISO repository from which VMs will be installed

If multiple operating systems will be used in a XenServer virtualisation infrastructure, it is efficient to set up an ISO repository or library. This repository prevents installion of VMs over the wire from the internet. It also prevents installation from physical CD and DVD media.

1. Launch the New Storage Repository wizard

From the XenCenter home screen, select the New Storage option to launch the New Storage Repository wizard.

2. Select ISO library type

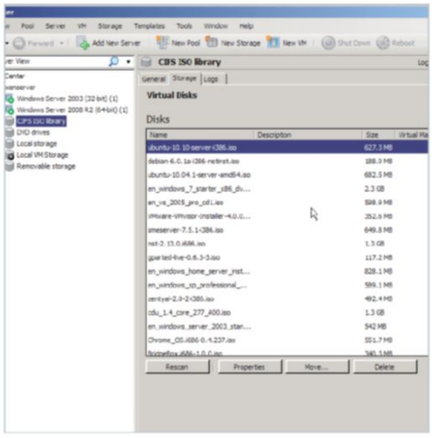
Select the type of ISO library from the two choices: Windows File Sharing (CIFS) or NFS ISO. A CIFS share does not have to originate from a Windows system. A Linux system running SAMBA will serve as a CIFS ISO library. Click Next to continue.

3. Connect to the shared resource

Name the CIFS library or accept the default name (CIFS ISO library) and enter the Share Name in the form: \\Server\\Share. If required, enter a username and password for the shared resource. Click Finish to complete the wizard and connect to the ISO repository.

4. Examine the ISO repository

The repository will appear on the XenCenter home screen's inventory list. Select the ISO library in the left pane. In the right pane, select the Storage tab to view the list of ISO files in the repository.



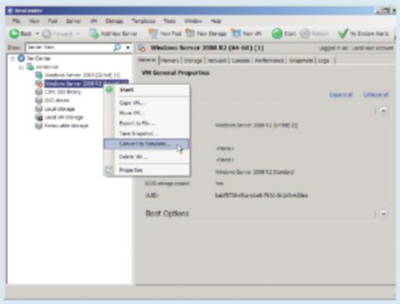
Creating a template from a virtual machine

Learn to create virtual machine templates as an agile resource

Virtual machine templates enable the rapid deployment of preconfigured virtual machines. Due to the convenience of this, templates are a valuable resource in virtualisation infrastructures.

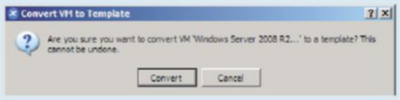
1. Select VM for template conversion

Choose a virtual machine to be converted to a template. The VM must be powered off prior to conversion. Select the powered-off VM, right-click, then select 'Convert to Template' to begin the conversion process.



2. Confirm VM conversion to template

Confirm the VM's conversion to a template by clicking the Convert button on the warning that appears as the first step.



3. Verify template conversion

The newly created template appears in the XenCenter inventory list. Note the change in the icon representing the template.

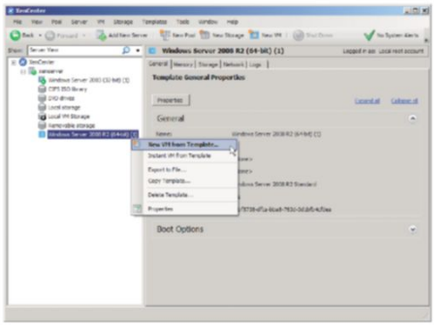
Deploying a VM from a template

How to deploy a new XenServer virtual machine from a template

Deploying a new virtual machine from a template is fast and easy. The additional space used by saving templates is offset by the efficiency of VM deployment.

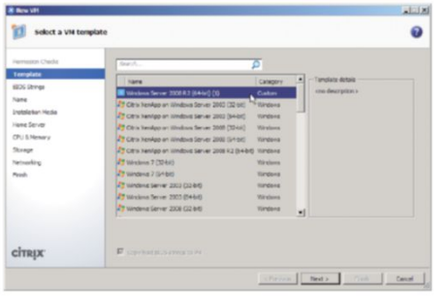
1. Start the New VM from Template wizard

Right-click the template that will be used for the new VM and select 'New VM from Template' to launch the deployment wizard.



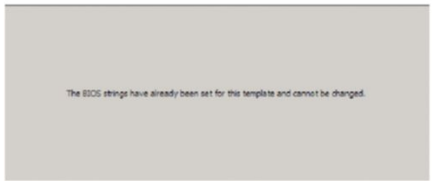
2. Select the VM template

Select the VM template from the list presented in this step. Click Next to continue.



3. Acknowledge notices

Click Next to acknowledge any warnings such as 'BIOS strings have already been set for this template and cannot be changed.' This is a nonfatal notification and can be ignored.



4. Complete the deployment wizard

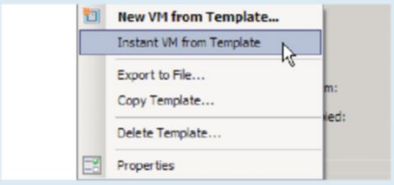
Step through the remainder of the 'New VM from Template' wizard. Do not set up installation media in the step that prompts for it. It isn't necessary to use installation media (ISO, CD, DVD, URL) when using a template. Once the wizard completes, the VM powers on and is ready for use.

Creating an instant VM from a template

Quickly create a new non-custom virtual machine from a template

1. Create an instant VM from template

Right-click the VM template from the XenCenter inventory list and select 'Instant VM from Template'. The VM is created in the background with no customisations.



2. Verify the creation of the new virtual machine

In a few minutes, the new VM appears in the inventory list, powers on and awaits a login.

Two-minute tutorials – Bash

Dive into Bash and take the mystery out of the Linux black box...

Welcome to the two-minute tutorial for Bash, the 'Bourne-Again shell'. Bash is the default Linux shell or command language interpreter. The command line is an efficient method of interacting with the Linux operating system. System administrators rarely install a graphical user interface (GNOME, KDE, LXDE) onto Linux server systems, therefore it is essential that everyone who takes Linux seriously learn how to navigate and use Bash.

The Linux shell is an interactive experience. The user enters commands, receives a response and enters new commands. It is a two-way conversation between the user and the operating system, with the shell acting as an interpreter.

Bash has two main components: utilities and a programming language. When one thinks of a Linux shell, thousands of cryptic commands may come to mind, but the utilities or commands included in Linux are useful and powerful components. Additionally, most of those utilities have a built-in help system. Bash is also a versatile programming language. Known widely as 'shell scripting', Bash programming offers the power of automating repetitive tasks and simplifying of complex commands.

These tutorials range from very easy to difficult. However, any user should be able to follow the logic upon seeing the output of the most complex bit of programming code.

Looking at Bash history

Don't bother typing when there is shell history to rely on

Using shell history is a fast way to repeat recently used commands or to search for commands that you have used in the more distant past.

01 Check the existence of the history file

Bash history resides in a hidden file in the user's home directory named .bash_history. Check that the file exists by listing all files.

```
khess@dante:~$ ls -la
total 32
drwxr-xr-x 3 khess khess 4096 2011-05-25 14:59 .
drwxr-xr-x 3 root  root  4096 2011-05-17 08:42 ..
-rw------- 1 khess khess  113 2011-05-25 14:53 .bash_history
-rw-r--r-- 1 khess khess  220 2011-05-17 08:42 .bash_logout
-rw-r--r-- 1 khess khess 3353 2011-05-17 08:42 .bashrc
drwx------ 2 khess khess 4096 2011-05-17 12:56 .cache
-rw-r--r-- 1 khess khess  675 2011-05-17 08:42 .profile
-rw-r--r-- 1 khess khess    0 2011-05-17 12:57 .sudo_as_admin_su
-rw------- 1 khess khess   51 2011-05-25 14:59 .Xauthority
khess@dante:~$
```

02 Examine the contents of the .bash_history file

List the contents of the .bash_history file using the cat command. The file lists recently used Bash commands using one line per command.

```
khess@dante:~$ cat .bash_history
sudo halt -p
sudo apt-get update
sudo apt-get upgrade
top
clear
sudo halt -p
ifconfig
sudo halt -p
ifconfig
exit
khess@dante:~$
```

03 Using arrow keys for Bash history

The keyboard's arrow keys (Up and Down) are the keys to Bash history. The Up arrow key pages back into Bash history and the Down arrow key pages forward. Press Enter to accept the command.

```
khess@dante:~$ grep sudo .bash_history
sudo halt -p
sudo apt-get update
sudo apt-get upgrade
sudo halt -p
sudo halt -p
khess@dante:~$
```

04 Searching history for a command

Use grep to search the .bash_history file for a command. The history file can grow very large and therefore it's often easier to search for a specific command.

05 Use the history command to look at history

Use the shell command 'history' to look at Bash history. The history command is a built-in Bash command, like the cd command, that cannot be used outside of the shell.

Placing processes in background

Use Linux shell multitasking magic to place lengthy processes in background

Having the ability to place processes in the background, while continuing work in the foreground, is but one advantage and beauty of Linux and its UNIX roots.

01 Creating a file system snapshot

To demonstrate how background processing works, create a snapshot of all files on the local file system. This process might take several minutes to complete. Use the command ' ls –lR / > files_list.txt' to create the list. Perform this task as the root user or use sudo. Note the amount of time this process takes.

02 Using ampersand to background a process

Create the same list of files, but add an ampersand (&) at the end of the command to place the command's process into the background. Once executed, the $ prompt returns and work may begin while the file system list operates in the background. Enter the command as 'sudo ls -R / > files_list1.txt &'.

03 Verify the file lists

Compare the two files to confirm that both exist and they contain a list of files from the local file system. The difference in size of the two files is small but comparable.

```
khess@dante:~$ ls -la
total 2720
drwxr-xr-x 3 khess khess    4096 2011-05-25 20:57 .
drwxr-xr-x 3 root  root     4096 2011-05-17 08:42 ..
-rw------- 1 khess khess     113 2011-05-25 14:53 .bash_history
-rw-r--r-- 1 khess khess     220 2011-05-17 08:42 .bash_logout
-rw-r--r-- 1 khess khess    3353 2011-05-17 08:42 .bashrc
drwx------ 2 khess khess    4096 2011-05-17 12:56 .cache
-rw-r--r-- 1 khess khess 1375462 2011-05-25 21:02 files_list1.txt
-rw-r--r-- 1 khess khess 1372650 2011-05-25 20:32 files_list.txt
-rw-r--r-- 1 khess khess     675 2011-05-17 08:42 .profile
-rw-r--r-- 1 khess khess       0 2011-05-17 12:57 .sudo_as_admin_successful
-rw------- 1 khess khess      51 2011-05-25 14:59 .Xauthority
khess@dante:~$
```

■ Listing file system snapshot files to compare sizes

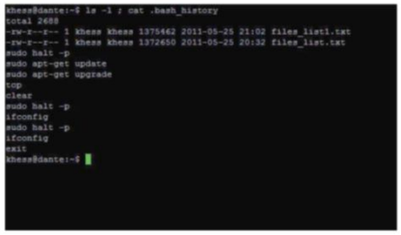
Running commands sequentially

Use the semicolon to stack commands and run them one after the other

Run commands sequentially by placing a semicolon (;) between each command in the format 'command ; command'. Use a single space between the command and semicolon for each command used. The semicolon runs the commands in the foreground, but without the need for user interaction.

01 Adding a semicolon

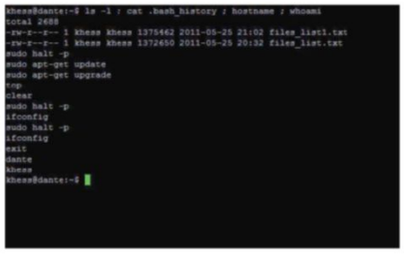
Place a semicolon between any two commands. For example, use 'ls -l ; cat .bash_history'. This command will display a long list of files in the current directory and then display the contents of the .bash_history file.



Stacking two commands using a semicolon

02 Running multiple commands sequentially using a semicolon

Using semicolons, run as many commands as needed – one stacked after the other. Each command will wait its turn before executing. The following command displays a long file listing, displays the contents of the .bash_history file, prints the hostname to the screen, and displays the username: 'ls -l ; cat .bash_history ; hostname ; whoami'.



Stacking four commands using the semicolon

> **A user can create a new file and add information into it simultaneously**

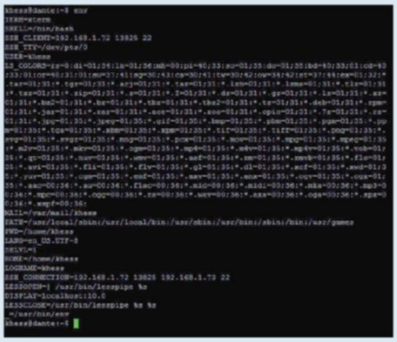
Using the echo command

The echo command displays information to stdout (the computer screen)

System administrators use the echo command to display environment information, to direct information into files, and to display return codes from the system.

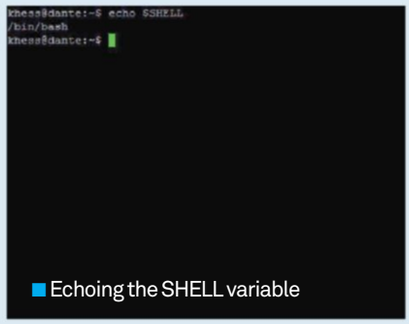
01 Display environment information

Upon logging into a Linux system, the user receives a shell environment that includes file execution paths, display redirection, language, shell, terminal and many others. To display all environment information, use the env command.



02 Echo user environment variable SHELL

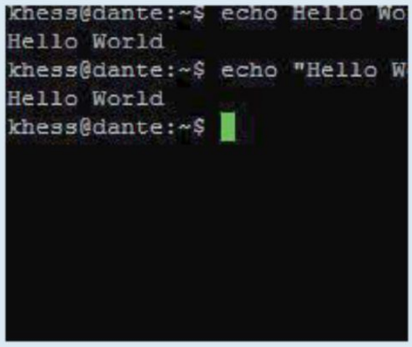
To echo any of the environment variables, use the echo command and the variable name with a preceding $ sign. The $ sign tells the system that the information to echo is a variable. Variable names are in ALL CAPS. Use the command 'echo $SHELL' to receive a response.



Echoing the SHELL variable

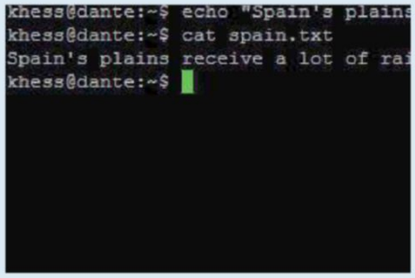
03 Use echo to send a message to the screen

Echoing a message to the screen (stdout) might seem superfluous, but it is not. It is a highly useful technique to use in script programming to send a message to the user. Quotation marks around the message are not required, but their use is considered good practice.



04 Writing information to a new file with echo

Using echo and shell redirection, a user can create a new file and add information into it simultaneously. Caution: If the file exists, using the > operator will overwrite the file and its contents with the new info. The command looks like the following: 'echo "Spain's plains receive a lot of rain." > spain.txt'.



05 Using echo to add information to a file

A user may echo information into an existing file without overwriting the file. Using the >> shell operator, echo adds information into a file. Use the following command to add a new line of information: 'echo "Is it rainy in Spain?" >> spain.txt'.

```
khess@dante:~$ echo "Is it rainy in Spain?" >> spain.txt
khess@dante:~$ cat spain.txt
Spain's plains receive a lot of rain.
Is it rainy in Spain?
khess@dante:~$
```

■ Adding new information to an existing file with echo

Looping as a first step to programming: the while loop

Learning to create shell programs is a fun way to interact with Bash

This example shows how to create a short but useful program to read a file, display its contents one line at a time and then exit.

01 Echo the .bash_history file contents line-by-line in a loop

This loop contains three essential components for a loop: Initiation (while), Action (do), Completion (done). Though looping programs can be far more complex than this one, they all follow the same basic structure. Enter the program at the command line: 'while read file ; do echo $file ; done < .bash_history'.

```
khess@dante:~$ while read file ; do echo $file ; done < .bash_history
sudo halt -p
sudo apt-get update
sudo apt-get upgrade
top
clear
sudo halt -p
ifconfig
sudo halt -p
ifconfig
exit
khess@dante:~$
```

```
khess@dante:~$ while read file ; do touch "$file".txt ; done < .bash_history
khess@dante:~$ ls -l
total 0
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 clear.txt
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 exit.txt
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 ifconfig.txt
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 sudo apt-get update.txt
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 sudo apt-get upgrade.txt
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 sudo halt -p.txt
-rw-r--r-- 1 khess khess 0 2011-05-28 16:10 top.txt
khess@dante:~$
```

Fig 1: Creating a list of files from .bash_history list

02 Creating a list of files with the loop

While echoing the contents of a file is infinitely interesting, a loop can do much more. The action in this script creates files from a list. Enter this command: 'while read file ; do touch "$file".txt ; done < .bash_history' (**Fig 1**).

```
khess@dante:~$ while read file ; do cat /etc/passwd > "$file".txt ; echo "$f
.txt ; done < .bash_history
sudo halt -p.txt
sudo apt-get update.txt
sudo apt-get upgrade.txt
top.txt
clear.txt
sudo halt -p.txt
ifconfig.txt
sudo halt -p.txt
ifconfig.txt
exit.txt
khess@dante:~$ ls -l
total 28
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 clear.txt
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 exit.txt
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 ifconfig.txt
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 sudo apt-get update.txt
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 sudo apt-get upgrade.txt
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 sudo halt -p.txt
-rw-r--r-- 1 khess khess 956 2011-05-28 20:15 top.txt
khess@dante:~$
```

Fig 2: Performing more than one action in a single loop

03 Create a list of files including data

A loop can have more than one action in it. This loop creates a list of files, cats data into each and displays the filename for each file. Enter the command: 'while read file ; do cat /etc/passwd > "$file".txt ; echo "$file".txt ; done < .bash_history' (**Fig 2**).

> "While echoing the contents of a file is infinitely interesting, a loop can do much more"

Creating a looping script from command line entries

Turn command-line success into a permanent script for repeated use

Save successful and useful command-line programs into files for reference, automation and efficient administration. Scripts can be reused and edited to apply to a range of repetitive tasks.

01 Starting the vi editor

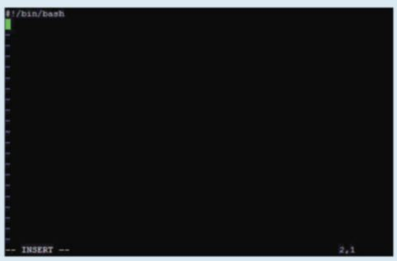
The first step in creating a new script is to open a text editor into which the script is written. Enter the following command to open a new file with the vi editor: 'vi create_files.sh'.

```
khess@dante:~$ vi create_files.sh
```

■ Invoking the vi editor

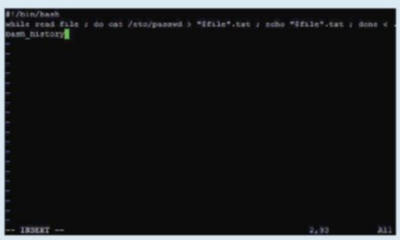
02 Set the script's execution shell

Once inside the vi editor, enter INSERT mode by pressing the I key. The word INSERT appears at the bottom of the editor screen. Enter the following execution shell setting: '#!/bin/bash'. This designation tells the system under which shell to execute the script.



03 Insert the loop from the previous how-to

Insert the finished command from step 3 of the previous how-to ('Looping as a first step to programming: the while loop' on the preceding page). Enter the command: 'while read file ; do cat /etc/passwd > "$file".txt ; echo "$file".txt ; done < .bash_history'. Copy and paste from the command line or rewrite the command as shown.



04 Remove command-line formatting

Press the Esc key once to enter COMMAND mode in vi. Use the arrow keys to navigate to the spaces and semicolons in the command-line script. Use the X key to delete spaces and semicolons. Use the I and Enter keys to move the remainder of the line down. Repeat the process until the script looks like the one shown in **Fig 3**.

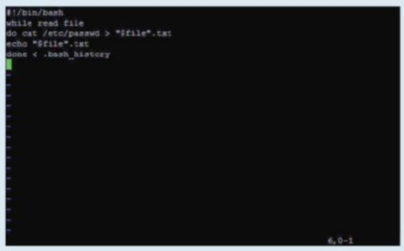


Fig 3: Command-line formatting removed

05 Saving the script

Press Esc to enter COMMAND mode in the vi editor. Hold the SHIFT key and press the "z" key twice to save and exit vi. Alternatively, press ESC, :wq, and ENTER. The script file is saved.

06 Changing script permissions to executable

A file is not a script, regardless of name, until it explicitly receives the execute permission. To do this, enter the command: 'chmod +x create_files.sh.'

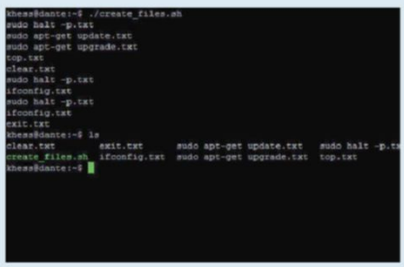


Fig 4: Executing the script file with the './create_files.sh' command

07 Executing the script

The script file may now be executed by using a special designation for scripts residing in the user's current directory. Execute a script with the (./)shorthand command. If a script is in a directory other than the user's current directory, use the full path to the script. Enter the following command to execute the script: './create_files.sh' (**Fig 4**).

> **A file is not a script, regardless of name, until it explicitly receives the execute permission**

Including a decision or condition in a script: IF-THEN-ELSE conditionals

Scripts become intelligent when they can make decisions based on predetermined criteria

Doing something in a loop is useful, but creating a script that can make decisions brings a new dimension to automation. The IF-THEN-ELSE conditional provides this functionality. If a condition is met, then some action(s) take place, else other action(s) take place.

01 Creating a decision point

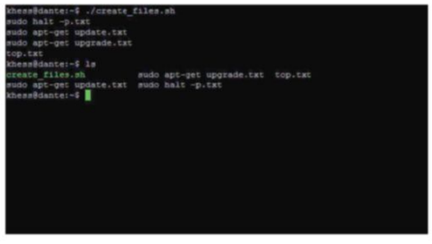
In this step, the decision to exit the script occurs when the $file variable equals "clear" from the .bash_history file. Until the script finds 'clear' in the list, the script will execute and create files. The script should create: sudo halt –p.txt, sudo apt-get update.txt, sudo apt-get upgrade.txt, and top.txt.

02 Editing the create_files.sh script

Open the create_files.sh script with the vi editor and enter the IF-THEN-ELSE conditional statement as shown.

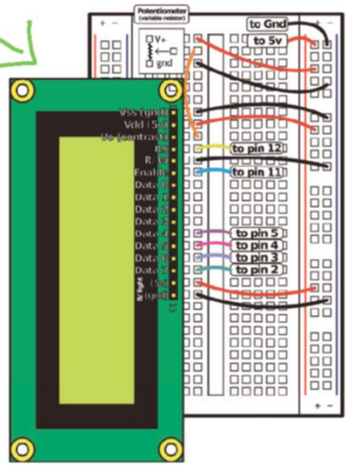
03 Executing the conditional script

Execute the script and observe the results. The script tests the conditions for each entry in .bash_history and creates files as instructed. Until $file = "clear", the script continues to execute. Once $file = "clear", the script exits gracefully.



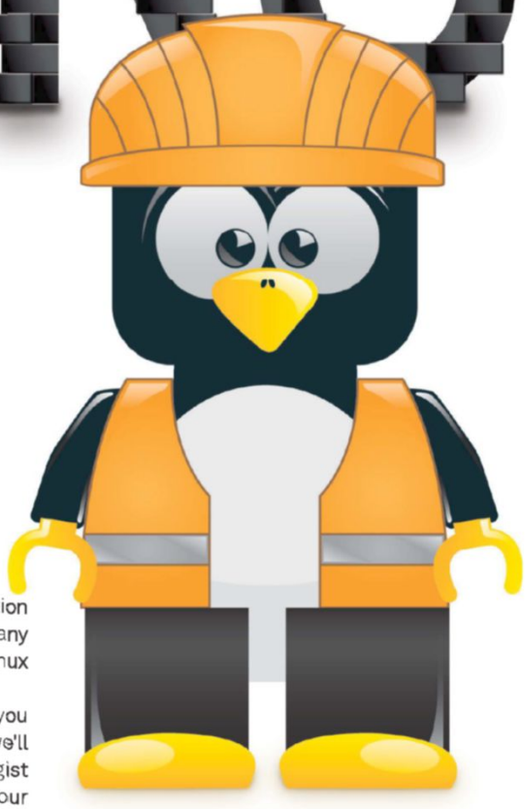
Linux Maste

Complete guides designed to help you advance your skills




rclass
php

BUILD YOUR OWN DISTRO



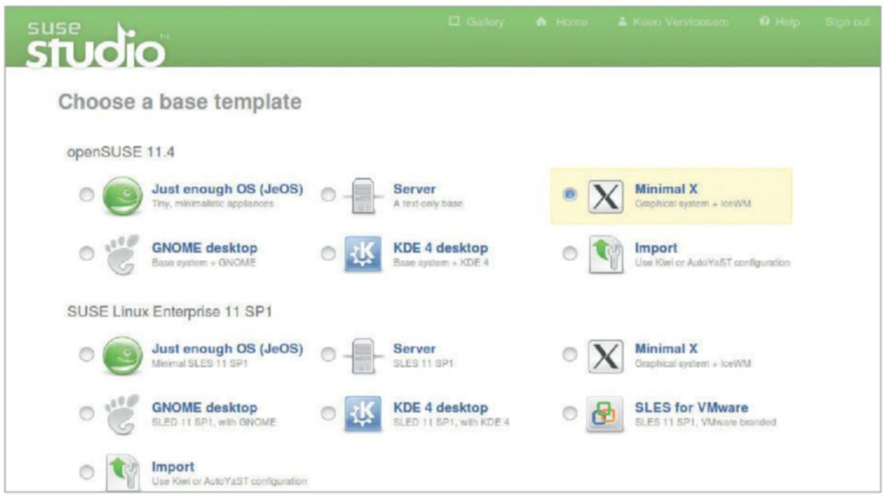
What if you don't like the default installation of any of the mainstream distributions? Don't panic, just create a distro yourself. Here's how...

If you're using one of the mainstream Linux distributions, you're forced to use the default desktop environment and applications that the distribution's developers have chosen for you. Of course, this is only an issue right after installation, because from then on you can add your own favourite software and delete packages you don't need, but this process takes time and is error-prone. If you have more than one Linux system, the problem is even worse: on each of these systems, after installation the distribution contains software you don't want or need. So these superfluous packages are downloaded and installed on every machine, and you have to customise it after installation every time, which is time-consuming and inconvenient.

That's where the idea of building your own distribution comes in. You take a base distribution and tailor it to your taste by adding and removing specific packages, adding your favourite configuration tweaks and so on. If you have a couple of Linux machines, this will save you a lot of time: you have to customise the installation just once while building your distribution, and after installing it on your various machines, you have the same set of your favourite applications and configuration on all your machines. But of course there are also other use cases: maybe you always wanted to create and distribute your own Linux distribution aimed at a specific niche. Or maybe you want to demo some application during a trade show or in class with a live CD, or you want a distribution with your company's branding. There are many possible reasons for building your own Linux distribution.

There are a lot of tools that can help you with building your own distribution, and we'll cover a couple of them, as well as the gist of some more manual ways to create your own distribution. First we'll take a look at SUSE Studio, a web-based and very user-friendly way to create your own SUSE-based distribution. Another web-based solution is Instalinux, which is not as user-friendly as SUSE Studio but supports more base distributions. After that, we cover some of Debian's and Ubuntu's distro builders, such as Remastersys and Ubuntu Customization Kit. If you're more of a Fedora fan, you can use Revisor. If you're a power user and you already have a customised Arch Linux distro on your hard drive, we'll show you how to produce an ISO image based on it. And last but not least, if you really want to build your distribution from the ground up, we'll show you the basics of Linux From Scratch.

Two years ago, Novell launched SUSE Studio, which it calls a "simple and fast appliance builder". It provides a free and easy-to-use, web-based user interface to roll your own customised (SUSE) Linux distro. Just create an account on **www.susestudio.com** and after the first login, the system offers some templates to build the appliance upon. You have the choice of openSUSE or SUSE Linux Enterprise for the base system, and there are templates for JeOS,



GNOME or KDE, server or desktop, it doesn't matter: SUSE Studio can build it all

SUSE Studio

Server, GNOME Desktop, KDE 4 Desktop and Minimal X – specially useful for making a kiosk-like distro.

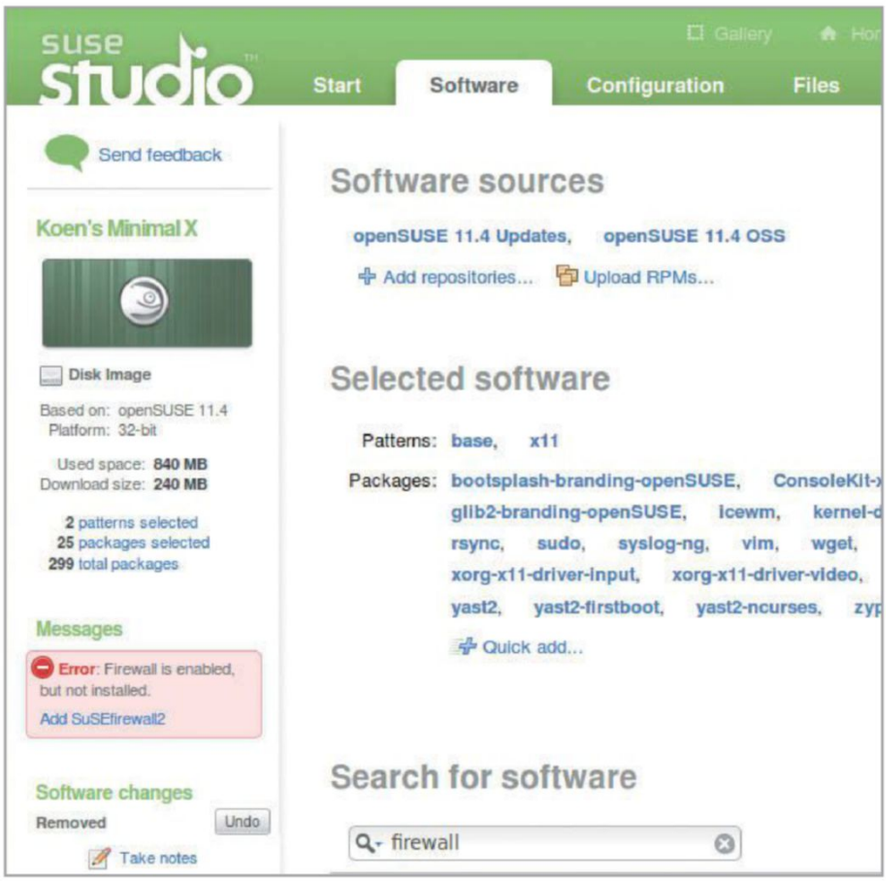
You navigate through the rest of the steps by clicking on tabs. In the Software tab, you can add or remove packages and repositories or even upload your own RPM files. This interface has a helpful search function. SUSE Studio is well engineered and has obviously undergone a lot of usability tweaking. For example, in the Software tab there's a 'Recommended' list of applications, which is different depending on the template you have chosen. Moreover, at any time the current disk footprint of the appliance is shown in a sidebar, which is useful while building an image which has to fit on a CD or a USB thumb drive. The sidebar also shows helpful messages and tips.

After the software choice, there are myriad other options to configure, like the network, firewall, users, the look and feel, a custom boot script etc. In the 'Overlay files' tab, for instance, you can add specific files to the root file system, such as custom scripts or configuration files in /etc. At the end, you can create an ISO image for a live CD, but SUSE Studio also supports many other image formats, including for Xen and Amazon EC2. The build process usually takes less than five minutes, even for a complete desktop system. After this, you can download the result.

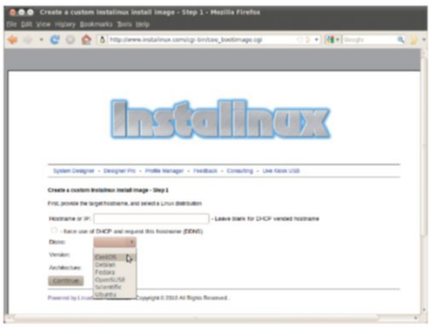
Pros
Very easy to use; very polished interface; web-based so no installation required; good documentation; Testdrive

Cons
Only SUSE-based distributions; downloading your distro could be slow depending on your internet connection speed

One great feature is Testdrive. This allows you to boot and test your distro in a browser window without the need to download it first. After the build, clicking the Testdrive button will launch the virtual machine on Novell's servers and expose its screen via VNC to a Flash applet running in your browser. There are even buttons to switch to a different virtual console, activate keyboard combos, and to change the keyboard layout. Each Testdrive instance gets 512MB of RAM and an hour running time on the server.



If you enable the firewall in the "Configuration" tab and it is not installed, then SUSE Studio shows an error message with a button to add SuSEfirewall2



Instalinux is not as polished as SUSE Studio, but at least it gives you more choice than only green distros

Instalinux

Pros
Web-based so no installation required; much choice for the base distribution

Cons
The interface is not polished; not very flexible; downloading your distro could be slow depending on your internet connection speed

If you like the web-based approach of SUSE Studio but don't want to commit to the SUSE branch of Linux distributions, Instalinux (on **www.instalinux.com**) could be a good solution for you. It's not remotely as polished as Novell's distro builder and it doesn't offer as many possibilities, but the big advantage of Instalinux is that you can build your own distro on top of a much wider set of base distributions: CentOS, Debian, Fedora, openSUSE, Scientific Linux or Ubuntu. And this both in 32-bit and 64-bit versions and for various releases.

After choosing the base image, you can select a proxy server which will be used by the ISO image as an FTP/HTTP proxy during install; you can choose the language, keyboard layout and time zone and so on. In the package selection step, you can select one of the base install profiles (eg ubuntu-desktop, kubuntu-desktop, or minimal for Ubuntu, or one of the many software bundles in CentOS), or you can place individual package names in a text box. In the last step, you enter the username, real name and password (which you obviously have to change after you have installed your distro). After clicking on 'Go For It!', Instalinux starts building your personalised distro and offers it soon as an ISO file. Download it, burn it to a CD and then start your computer from the CD. On the boot prompt, type 'install' to automatically install the system with your chosen configuration.

Ubuntu Customization Kit

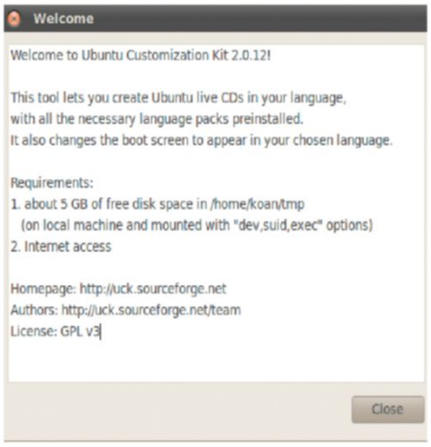
Pros
Very easy to use

Cons
Only Ubuntu as a base system; base system needs to have the same version and processor architecture as the OS you're running; no easy way to customise the look and feel; you have to start over from scratch when you want to change your customised distribution

The easiest way to create an Ubuntu-based distribution is Ubuntu Customization Kit. Download an existing Ubuntu ISO image, install Ubuntu Customization Kit (UCK) from your package manager and make sure you have 5GB free space in your home directory. Then launch Ubuntu Customization Kit and follow the instructions in the subsequent dialog screens. They provide you with everything you need to customise the default Ubuntu ISO image.

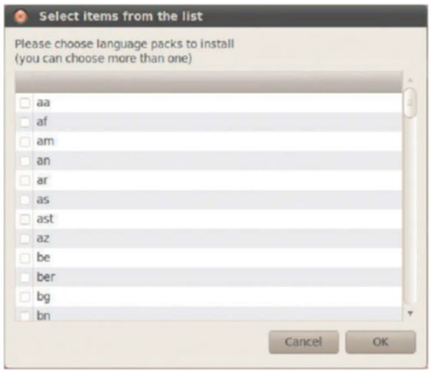
01 Requirements

The first screen welcomes you and reminds you about the requirements: be sure to have 5GB of free disk space in your home directory and internet access. For more information, the welcome screen refers you to the homepage of Ubuntu Customization Kit (**http://uck.sourceforge.net/**).



02 Language packs

In the next screen you choose the language packs to install, at least if you want to use another language than English. By default, the installer of the Ubuntu live CD supports all languages, but the other programs only have English language files. In the next screen you can choose the available languages in GRUB's boot menu; and in the screen after it, the default language.



03 ISO file

If you're just using English, you can ignore the previous screens, but the next screen is really important. You have to choose an ISO image to base your distro on. This could be a plain Ubuntu image, or a Kubuntu or Xubuntu one. One thing that you really should take care of: the image should be of the same Ubuntu version as the Ubuntu distro you are running UCK on, and the same processor architecture (32- or 64-bit).

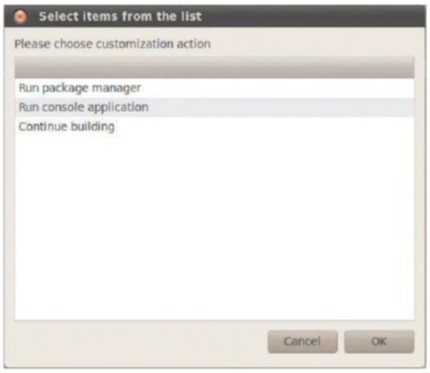
04 Name

Now enter the name of your personalized distribution. This is not so important, but it's convenient if you want to recognise your distro if you put the CD or USB stick in your computer: your distro's favourite file manager shows this name with the icon of the CD or USB stick.

05 Manual customisation

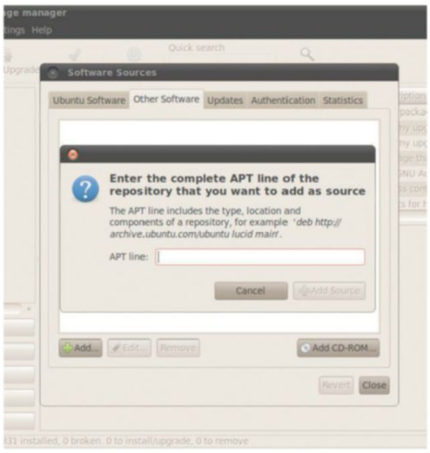
In the next step, UCK asks you if you want to customise the ISO image further

manually. If you answer No, you just end up with a default Ubuntu ISO image with minor differences, such as the language and the CD's name. If you answer Yes, UCK will offer to open Synaptic or a Terminal window to start your customisation, but not after first unpacking the ISO.



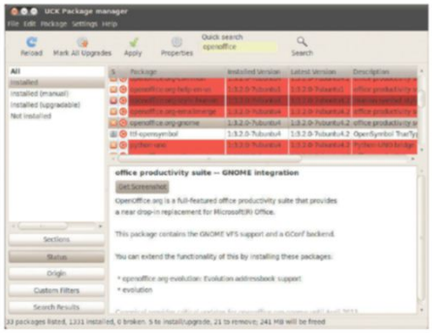
06 Repositories

UCK starts Synaptic. In its settings, you can activate repositories like universe or multiverse or even add third-party repositories like Medibuntu. Following that, click on Reload to update the package list, after which you can install your favourite packages and remove packages you don't need. For the latter, be sure to choose 'Mark for complete removal'.



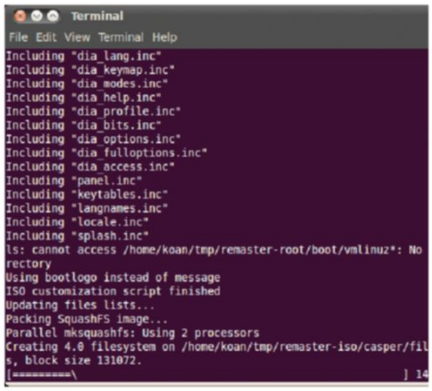
07 Packages

We removed OpenOffice.org, Firefox and other bloated software, and added lightweight alternatives like AbiWord, Gnumeric and Chromium. When you're done with your package choice, just click on Apply and quit Synaptic, which brings you back to the previous UCK screen. Now you can choose to do some manual customisations in a Terminal window or click on 'Continue building'.



08 Build

After you have applied all your customisations, UCK starts building your personalised distribution. You can follow the output of the process in a Terminal window. After 10 to 15 minutes, your personalised ISO image is ready. Now you can test it in VirtualBox or KVM and if it seems to work, you can burn the ISO image to a CD-ROM or write it to a USB stick with UNetbootin or the Ubuntu USB image installer.



"The easiest way to create an Ubuntu-based distro is Ubuntu Customization Kit (UCK)"

Bodhi Linux

Jeff Hoogland created Bodhi Linux because he wanted a Linux distribution that ships a current Enlightenment desktop

■ Jeff Hoogland, creator of Bodhi Linux

So, which customisations did you apply in Bodhi Linux?

We do our best to make things easy for those new to Enlightenment to be able to use. Beyond this, a large amount of the software you see in Bodhi comes down from the Bodhi repository as opposed to the Ubuntu repository. This way we can deliver the latest, stable software to our users.

Which tools did you use to build Bodhi?

When building the Bodhi ISO, I started with an Ubuntu 10.04 minimal ISO. I then added the Bodhi repository and downloaded the rest of the system packages and set up the Bodhi configurations. I did this all in a VirtualBox image, so I could easily roll the system back to various snapshots should I decide to revert a change I have made. As for building the Bodhi ISO image itself, I am a fan of the Remastersys tool. It is simple and works well for the task it was designed for.

Do you have any tips for people who want to build their own Linux distro?

Before you start, stop and think to yourself: Is this truly going to be useful? What can you do with a Linux distribution that hasn't already been done? Why should people use what you are going to create instead of other Linux distributions out there?

As far as getting your distribution out there, know from the start you are not going to please everyone. A good ability to have, however, is being able to sort out the people that just like to whine, from those that are actually providing constructive criticism. And finally, don't be afraid to ask for help. Building and maintaining a Linux distribution is not (and should not be) a one-man show.

Remastersys

Pros
Very flexible; fast way to copy your complete working environment

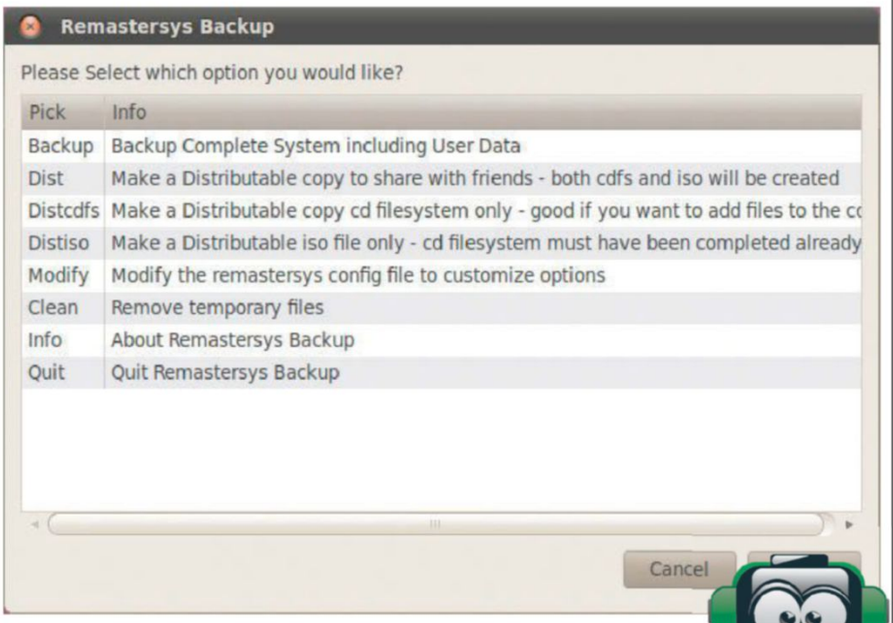
Cons
Only Debian or Ubuntu as a base system; not yet in Ubuntu's official repositories; you need a separate Debian/Ubuntu installation to base your image on

While Ubuntu Customization Kit lets you customise an existing Ubuntu ISO image, Remastersys (which supports Debian and Ubuntu) just creates an ISO image based on your current Ubuntu installation on your hard drive. You can do this in two ways: by creating a backup or a distro. The backup option creates a complete backup of your system, including your home directories. So this is a fast way to be able to restore your complete working environment after a disaster, or to migrate your exact working environment to another computer.

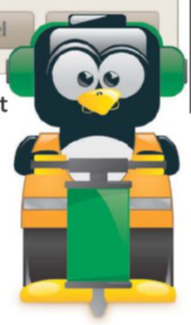
The other option is more interesting: to build your own distribution. This creates a full backup of your system, but without the home directories, so your personal files and settings are not put in the ISO image. However, you can customise personal settings, as we'll see soon. The first thing you should do before creating a distro image with Remastersys is to have a clean system with your customisations. You can do this on your own computer, but better is to install a clean system on a virtual machine, eg in VirtualBox. Update the repositories and upgrade all packages to their latest version with 'sudo apt-get update' and 'sudo apt-get upgrade', and remove older kernel versions.

Now install and remove packages at will in this machine. After this, clean the package cache with 'sudo apt-get clean' and wipe all temporary files with 'sudo rm -rf /tmp/*'. If you want to apply personal settings to the home directories of the users in your personalised distro, you have to copy the right configuration files from your own home directory on the machine to /etc/skel. For instance, if you install the system monitor Conky, you can install a customised .conkyrc in /etc/skel. Or you can add things to .bashrc or other configuration files. When users of your distro install it and define a user, the Ubuntu installer copies all files from /etc/skel to the home directory of the newly created user. The only thing you shouldn't forget is setting the right permissions after copying your configuration files to /etc/skel with 'sudo chown -R root:root /etc/skel'.

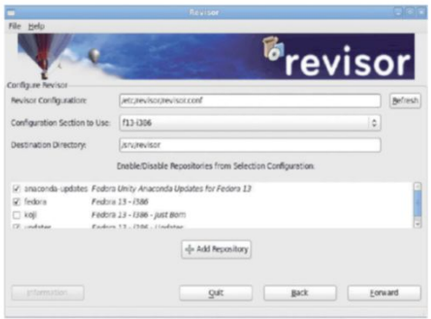
To install Remastersys, first add the right repository to /etc/apt/sources.list (consult the Remastersys website on **http://geekconnection.org/remastersys/** for this). After this, you can find it in the GNOME menu System>Administration. Choose the right option in the program, after which the distro creation starts immediately. You can follow the progress in a Terminal window.



Remastersys is the fastest way to copy your complete working environment



"The most geeky way of building your own Linux distro is definitely Linux From Scratch"



Revisor is a user-friendly wizard to create your own Fedora spin

Revisor

Pros
Easy to use; many possibilities in the wizard interface

Cons
Only Fedora as a base system

If your favourite distribution is Fedora, you're not out of luck. There's a tool called Revisor which has roughly the same approach as Ubuntu Customization Kit. A difference is that it doesn't require you to download an ISO image first: Revisor automatically downloads the required files to create a new ISO image of the Fedora-based distro you're building. You can find more information on its website, **http://revisor.fedoraunity.org/**.

Revisor has a wizard-like interface which walks you through the steps of creating your ISO image. After the welcome screen, you choose the media type (installation or live) and in the next screen you select the repositories you want and you can add third-party repositories. In this screen you can save these settings as a Revisor configuration. After the repositories are loaded, you can add or remove packages at will. When you have chosen your package set, there are some final steps with the default language, keyboard type, time

zone, network interface configuration and so on. There's even a screen that sets up a firewall and another one to configure X, including which desktop environment to run. After adding a user, click on Forward to start building your own Fedora spin.

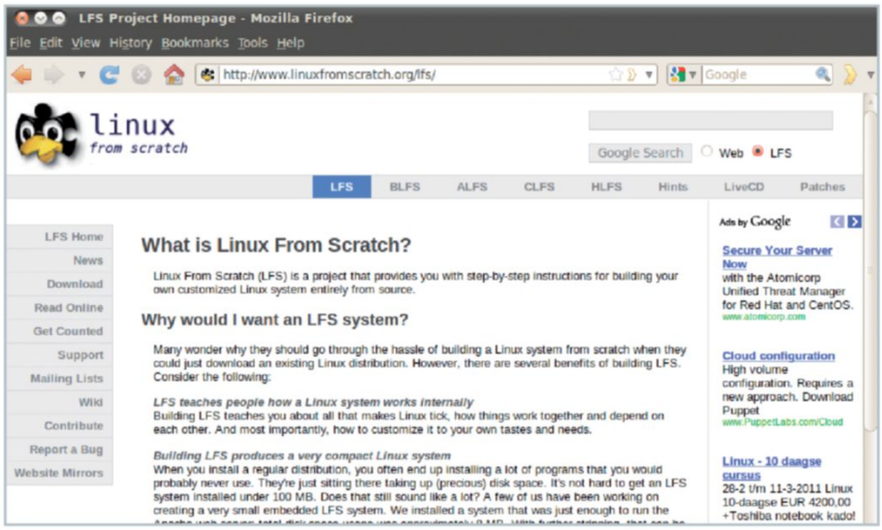
Larch

Pros
Flexible; good documentation; builds fresh images or ones based on an existing Arch Linux install

Cons
Only to create Arch Linux derivatives; not for beginners (just like Arch Linux)

Arch Linux is by design intended to be lightweight and customisable, so it's the perfect building block for your own distro. But how do you do this? You can just install Arch Linux and start customising it, but then this still requires a lot of manual steps. Luckily, there's the Larch project, a live CD construction kit for Arch Linux. With Larch, you can produce a custom Arch Linux installation CD.

Larch itself doesn't require Arch Linux: it should be able to run on any Linux distribution, as long as you have some basic tools installed. Larch provides command-line scripts, but it also has a GUI. You can choose to build a fresh image based on an example profile or you can build it based on your existing Arch Linux installation. You can create your own profile and then you can list additional packages in a configuration file, each package name on a new line. Boot options, repositories, the locale and so on can all be configured in the profile.



Use the source, Luke! Linux From Scratch is the most profound way to build your own distro

Linux From Scratch

The most geeky way of building your own Linux distro is definitely Linux From Scratch (LFS). Go to the website **www.linuxfromscratch.org** and read the book for step-by-step instructions on how to build a complete Linux distribution starting from nothing but the source code. To start building LFS, you need a working Linux system with a compiler and some other development tools, and a clean partition to install LFS on. The first thing you do is compile a 'toolchain' with GCC, glibc, binutils and other basic development tools. Then you chroot to the toolchain's partition and start building your LFS distro there.

Pros
Extremely flexible; the result is compact, flexible and secure

Cons
Requires a lot of manual steps and quite some understanding about how a Linux distro works

LFS is really a small system, probably not that useful in many circumstances. However, the LFS developers have compiled a lot of further instructions in the online book, Beyond Linux From Scratch (BLFS). For instance, this teaches you how to configure networking, X and devices such as printers, scanners and sound. It also presents a couple of choices for the desktop environment and window manager, and shows you how to get some common desktop applications up and running. And if you want to build a really secure Linux distro, you should consult the Hardened Linux From Scratch documentation. But all of this requires manual steps from you, so to really build a distro you should consult the Automated Linux From Scratch project.

All in all, Linux From Scratch is the most powerful way to build your own distro, but it's not that easy and requires you to make a lot of decisions yourself. You have complete control over the way that the packages are built, which patches are applied and so on, which is extremely flexible, but with great power comes great responsibility: with LFS you should know what you're doing.

Zorin OS

Kyrill Zorin and his team created Zorin OS to help Windows users to enjoy the benefits of Linux

Kyrill Zorin, co-founder of Zorin OS and Zorin PC

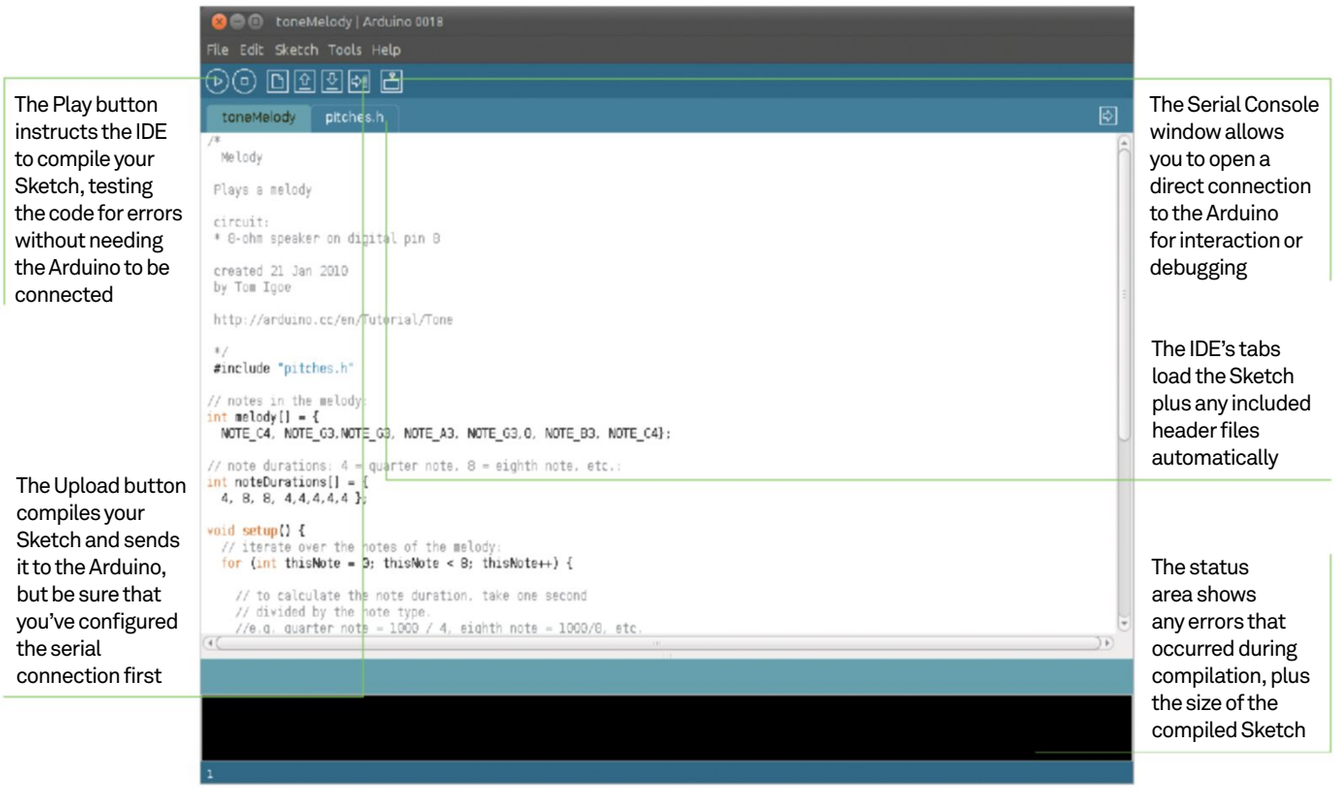
Which customisations did you apply in Zorin OS?
We developed the Zorin Look Changer which allows users to switch between different GUIs, including Windows 7, Vista, XP, 2000, Mac OS X and GNOME. Our team has developed the Zorin Splash Screen Manager which makes it easy to customise the Plymouth splash screen, and Zorin Background Plus which adds support for animated wallpapers. We have included media codecs out-of-the-box and made other improvements to simplify the setup for new users.

Which tools did you use to build Zorin OS?
We used Remastersys and ISOLINUX to create Zorin OS.

Do you have any tips for people who want to build their own Linux distro?
We would advise new developers to find their niche so that their distro would stand out rather than replicate other distros. Alternatively they can join existing distro teams and contribute their new ideas and features to that distro.

Developing with Arduino – part 1

Gareth Halfacree shows us how to get started with open hardware and create an electronic music box with no prior experience necessary...



Advisor

Gareth Halfacree has been breaking, fixing, tinkering, and generally voiding the warranty on his electrical and electronic items for years, and hasn't received a fatal electric shock yet

The Arduino project is a hacker's dream: originally designed as a way of getting students involved in microcontroller programming and electronic design, it offers a quick and easy route for programmers to take their skills out of the digital realm and into the physical.

Better yet, the Arduino's plug and play nature and the C-based programming environment mean that it's easy to get started, even if you lack experience in electronics.

> "It's easy to get started, even if you lack experience in electronics"

To prove it, this three-part project is going to walk you through building a simple Arduino music box, programming it with the Arduino IDE, integrating it with your PC, and finally adding a useful LCD display to your project.

As a completely open source project, Arduino programming goes well with Linux development, and all the code and hardware designs behind Arduino are freely available for tinkering.

Before we can get started, we'll want to install the Arduino development environment. If you're lucky enough to be running a version of Linux for which pre-packaged versions exist, it's a simple case of downloading them from the Arduino website or installing directly from your distribution's repositories. If not, there are a few steps we'll have to go through before we can get started with building our first Arduino circuit.

> Arduino is a hacker's dream

Installing the Arduino IDE

1. Grab the dependencies

If you're not using a pre-packaged version of the Arduino IDE, you'll need to install certain dependencies. **In Debian-based distributions, simply run:**

```
$ apt-get install openjdk-6-jre gcc-avr avr-libc avrdude binutils-avr
```

For Fedora-based distributions, the dependency list is slightly different:

```
$ yum install uisp java-1.6.0-openjdk avr-libc avr-gcc-c++ rxtx avrdude
```

2. Change group membership

As the Arduino IDE requires access to the USB port to program the board, you'll need to make sure that you're a member of the right groups. **As root, change the group memberships as follows:**

```
$ usermod -a -G uucp,dialout,lock $USER
```

3. Install Arduino

The easiest way to get the Arduino IDE running on your system in lieu of a suitable package is to download the pre-compiled binaries and extract them into your home directory:

```
$ wget http://arduino.googlecode.com/files/arduino-0021.tgz && tar xvfz arduino-0021.tgz
```

If you're running a different distribution, or if the steps outlined above don't work for you, more detailed instruction are available on the Arduino Playground at http://www.arduino.cc/playground/Learning/Linux.

Resources

Arduino Experimenter's Starter Kit
http://oomlout.co.uk or

Arduino, breadboard and wiring, piezoelectric buzzer
http://oomlout.co.uk

Arduino IDE http://arduino.cc

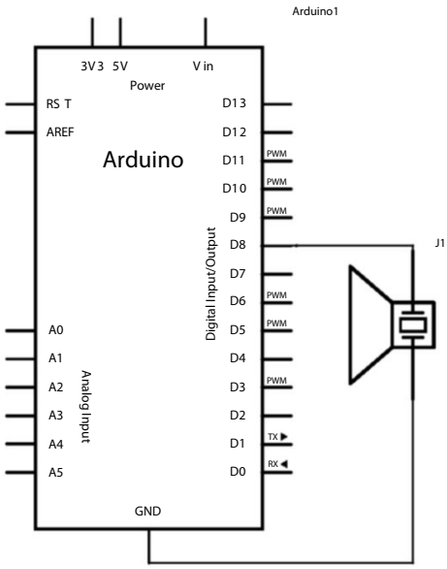


Fig 1 Connect the piezoelectric buzzer to Pin 8 The program will control the buzzer through Pin 8 on the Arduino

Wiring up your first circuit

Before we get started writing our project in the Arduino IDE, we need to build the circuit. For that, we'll need our Arduino development board, a piezoelectric buzzer, some wire and a breadboard. If you're using the Arduino ARDX Kit from Oomlout, then you've already got all the parts required.

The Arduino board features multiple pins, some of which are capable of acting as analogue inputs for sensors while others can act as pseudo-analogue outputs thanks to a technique known as 'pulse width modulation'.

For our simple starter project, however, we're only going to use a single digital output pin.

4. Connect the piezoelectric buzzer to Pin 8

Our program is going to control the buzzer through Pin 8, one of the Arduino's digital output pins. Before that can happen, we need to connect the buzzer to the output pin on the breadboard (**Fig 1**). If you're using wire instead of a breadboard, just connect the positive wire directly to the pin.

5. Connect the piezoelectric buzzer to the Ground Pin

Before we can use the buzzer, it needs to have a path to ground in order to complete the circuit. Thankfully, the Arduino includes a ground pin that we can connect directly into (**Fig 2**).

6. Check your circuit

The finished circuit should connect the buzzer to the Arduino, and look something like the picture on the next page (**Fig 3**). Double-check that the wires lead to the same row as the buzzer's pins, as it's easy to miss.

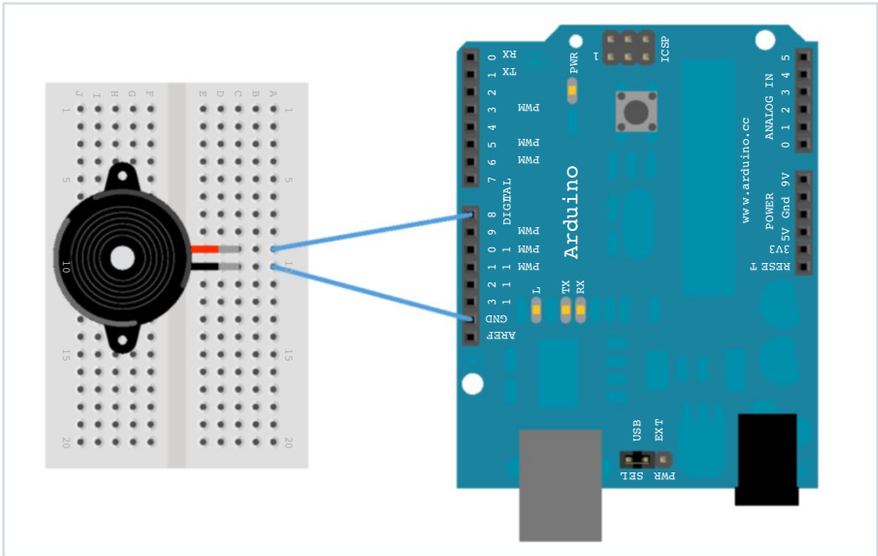


Fig 2 Connect the piezoelectric buzzer to the Ground Pin Completing the circuit

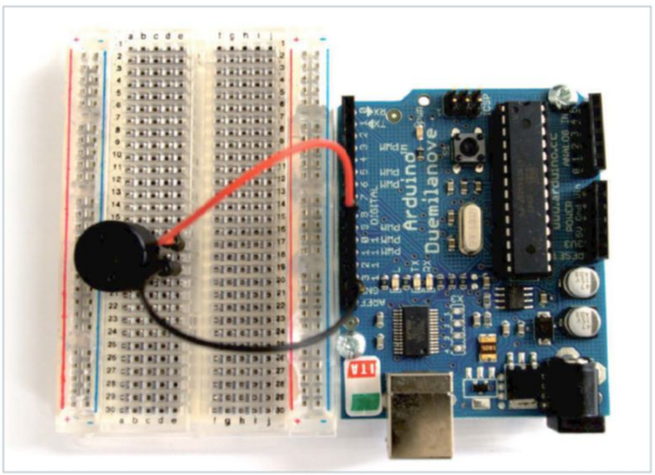


Fig 3 Check your circuit The finished circuit should look something like this

> **The complexities of coding for a microcontroller are abstracted away thanks to a library called Wiring**

Programming the music box

As it stands at the moment, we've got a simple circuit that doesn't do anything. In order to make our collection of components become the electronic music box that we originally envisaged, we need to tell the microcontroller at the heart of the Arduino exactly what we have plugged into it and how we want it to operate the components.

Thankfully, the processing-based Arduino IDE makes this easy. The complexities of coding for a microcontroller are abstracted away thanks to a library called Wiring, which allows you to address each pin of the microcontroller in a simple manner.

Anyone familiar with the C programming language will feel immediately at home writing for the Arduino, but there are some 'gotchas' that can trap the unwary.

An Arduino program is known as a 'Sketch,' and to make our simple music box we're going to load one of the example Sketches to customise rather than reinventing the wheel.

7. Load the sample Sketch

After loading the IDE, you can find the Sketch we want – written for the Arduino project by Tom Igoe back in January – under:

`File>Examples>Digital>toneMelody`

This Sketch contains an included header file, called 'pitches.h', which will also load automatically (**Fig 4**).

8. The empty loop

Usually the meat of a Sketch can be found in the loop section, which runs continuously while power is applied to the Arduino. Because a constantly repeating tune would be annoying, you'll notice that this section is empty in our sample program.

```
void loop() {
  // no need to repeat the melody.
}
```

9. The setup

Ordinarily, the setup section of a sketch contains the information needed to configure the Arduino for our commands in loop. However, because anything in setup is only executed once, it's where the meat of our single-play music box can be found.

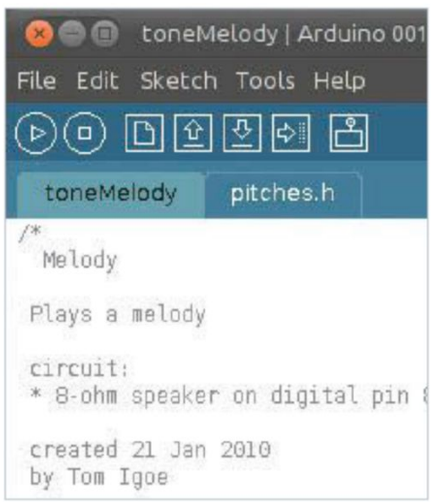


Fig 4 Load the sample Sketch Load up the toneMelody Sketch to customise

```
void setup() {
  // iterate over the notes of the
melody:
  for (int thisNote = 0; thisNote <
8; thisNote++) {

   // to calculate the note
duration, take one second
   // divided by the note type.
   //e.g. quarter note = 1000 / 4,
eighth note = 1000/8, etc.
   int noteDuration = 1000/
noteDurations[thisNote];
   tone(8, melody[thisNote],noteDura
tion);

   // to distinguish the notes, set
a minimum time between them.
   // the note's duration + 30%
seems to work well:
   int pauseBetweenNotes =
noteDuration * 1.30;
   delay(pauseBetweenNotes);
  }
}
```

10. Programming the Arduino

Before we start playing with the sample code, let's check that it runs by hitting the Verify button in the top-left of the IDE – the one that looks like a play symbol. All being well, you should receive a message telling you the compiled size of the Sketch at the bottom of the IDE. If not, you'll see an error message; the most common problem at this point is an inability to find the pitches.h file.

Now we know that the program is valid, let's put it into our Arduino. Connect your USB cable to the Arduino (**Fig 5**) and hit the penultimate Upload button. After a brief pause and a flashing light on the Arduino, you should be rewarded with an enchanting ditty from your very own music box.

Troubleshooting

If you didn't hear any music when you uploaded the Sketch to the Arduino, something's gone wrong. Don't panic, because there's usually a simple explanation. Let's have a look at what you may have missed.

11. Error messages

The most common cause of a failed upload is having the wrong serial port configured in the IDE, which will generate an error in the status area at the bottom of the IDE. Click the Tools>Serial Port menu and see what's set. If it's currently configured to /dev/usb0, try changing it to /dev/usb1 and clicking Upload again.

12. The circuit

Make sure that the buzzer is connected to both Pin 8 and Ground on the Arduino. If you're using a breadboard, it's easy to miss the right holes due to the size of the buzzer, so lift it and double-check it's connected to the same rows as your positive and ground wires (**Fig 6**).

13. Pitches.h

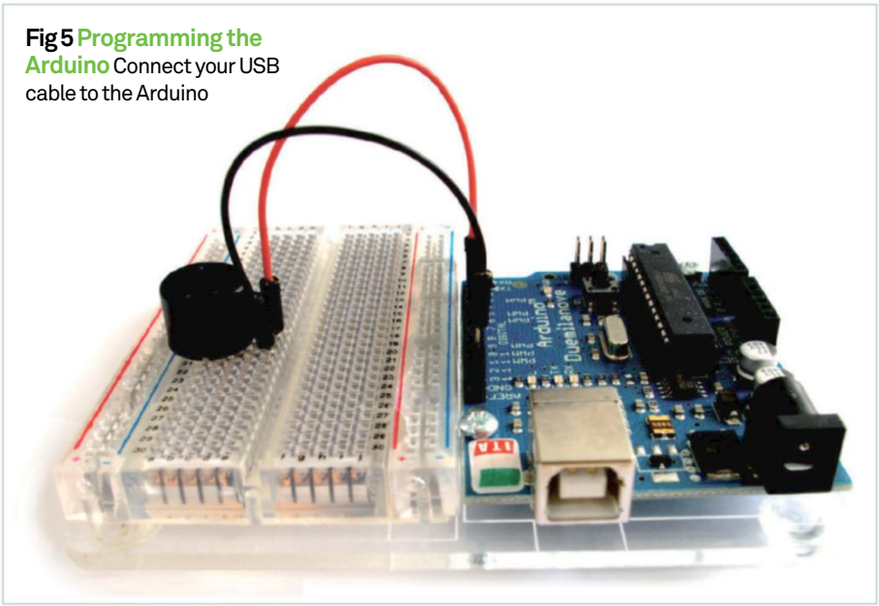
If you compiled the Arduino package yourself, or used an unofficial package, it's possible you don't have the file pitches.h in a location that the compiler can find it. If so, you'll see a 'No such file or directory' error in the status area. Try downloading a newer build, or find the missing file:

```
$ locate pitches.h
```

14. Customisation

Now that we've got our music box playing, it's time to try changing the Sketch to suit our own needs. Although the example is useful, the layout can be a bit strange and requires some careful handling if we want to play our own music.

Fig 5 Programming the Arduino Connect your USB cable to the Arduino



15. The pitch

To change the tune, find the melody section that defines what notes are played. Each note is defined as a tone from pitches.h. If you want to change the pitch of a note, do so here – but if you want to add more notes, there are a few more steps. A '0' represents a rest.

```
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4};
```

16. The length

Each note in our tune has a length value associated with it, including the rests. A value of 8 represents an eighth, 4 a quarter, 2 a half and 1 a full-length note. If you add more notes to the tune, be sure to add their lengths to this list.

```
int noteDurations[] = {
  4, 8, 8, 4,4,4,4,4 };
```

17. All together now

The last stage of adding to our tune is to delve into the setup section and change the line that iterates through each note. If we leave it at its default value, we'll only hear the first eight notes. Simply change the number in bold to the number of notes in your new tune.

```
void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {
```

Once you've changed your tune, hit the Upload button again to replace the old version of the Sketch still stored in the Arduino's memory with the new version. If you want to hear the tune again without re-uploading it, just hit the reset button on the Arduino's body.

Next month we'll look at tidying the example program up to make it easier to modify, and interfacing our music box with our Linux system for some interactive entertainment...

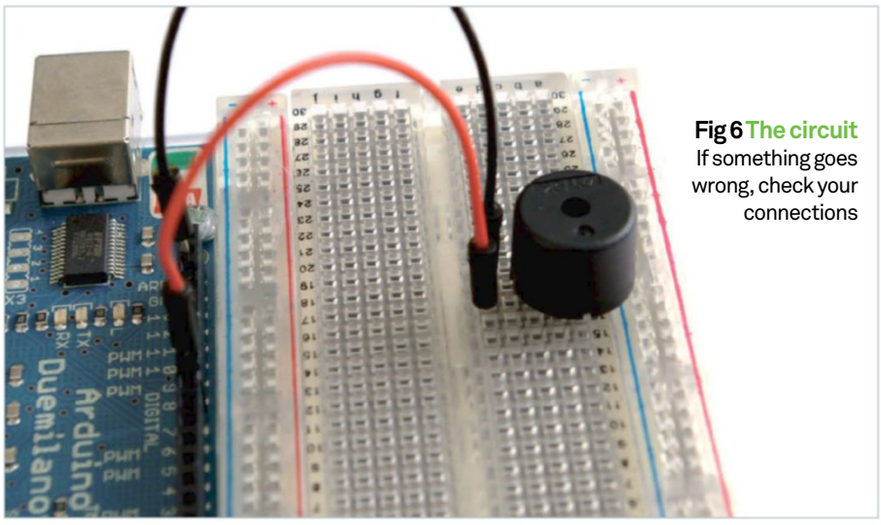


Fig 6 The circuit If something goes wrong, check your connections

Developing with Arduino - part 2

Building on the previous project, we delve into the world of Arduino web connectivity and produce something a little more useful...

Resources

Arduino Experimenter's Starter Kit
http://oomlout.co.uk or

Arduino, breadboard and wiring, piezoelectric buzzer
http://oomlout.co.uk

Arduino IDE http://arduino.cc

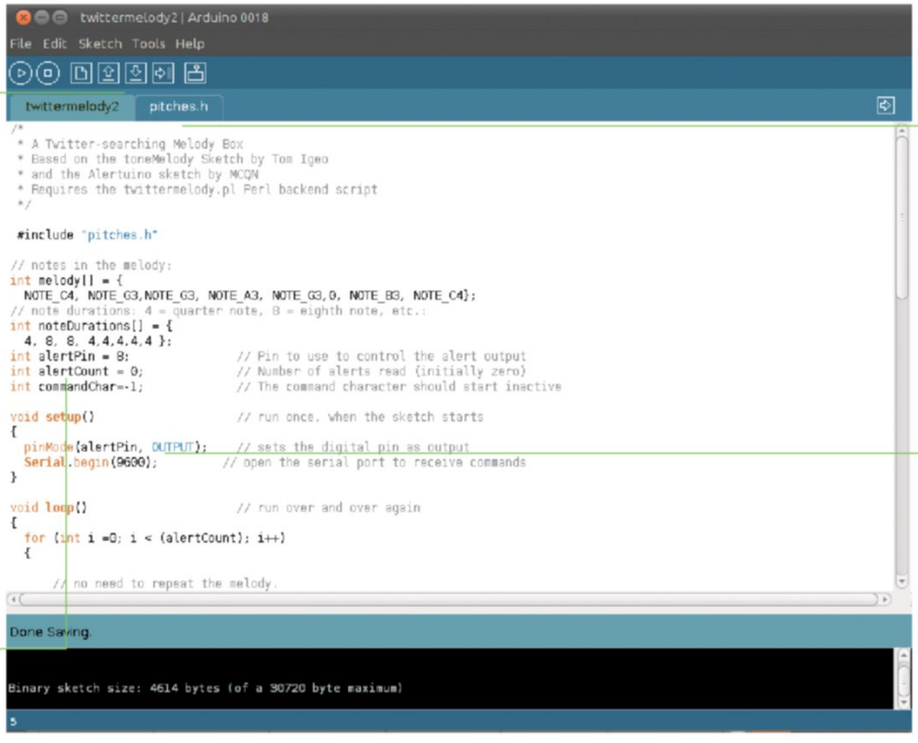
If you've ever wanted to create your own electronic equipment, but never knew how, the Arduino project is for you: originally developed as a teaching aid, it allows people with little or no electronics experience to build their own circuits and program the integral microcontroller with ease.

Building on our Arduino-powered music box of last month, we're going to take a look at how the Arduino platform can interface with a program running on your Linux-based PC. Rather than just playing the music once when the Arduino is first powered up, we're going to add a back-end package – written in Perl – that searches the popular microblogging service Twitter for a hashtag of our choice and plays a tune each time it's mentioned, demonstrating the Arduino's ability to accept input from external sources and take action based on what it receives.

If you missed last month's tutorial, or you've broken down the simple music-box circuit already, we'll start with a quick recap as to the circuit you'll need for this project.

Remember to upload the Sketch to the Arduino every time you make a change to how it works

New variables allow us to count the number of times the melody generator should run based on Twitter traffic



As our new Sketch builds on the old melody box, it needs the same header file

A serial connection to the PC allows the Sketch to receive data from a Perl-based Twitter search tool.

> "How Arduino can interface with a program on your Linux-based PC"

Advisor

Gareth Halfacree has been breaking, fixing, tinkering and generally voiding the warranty on his electrical and electronic items for years, and hasn't received a fatal electric shock yet

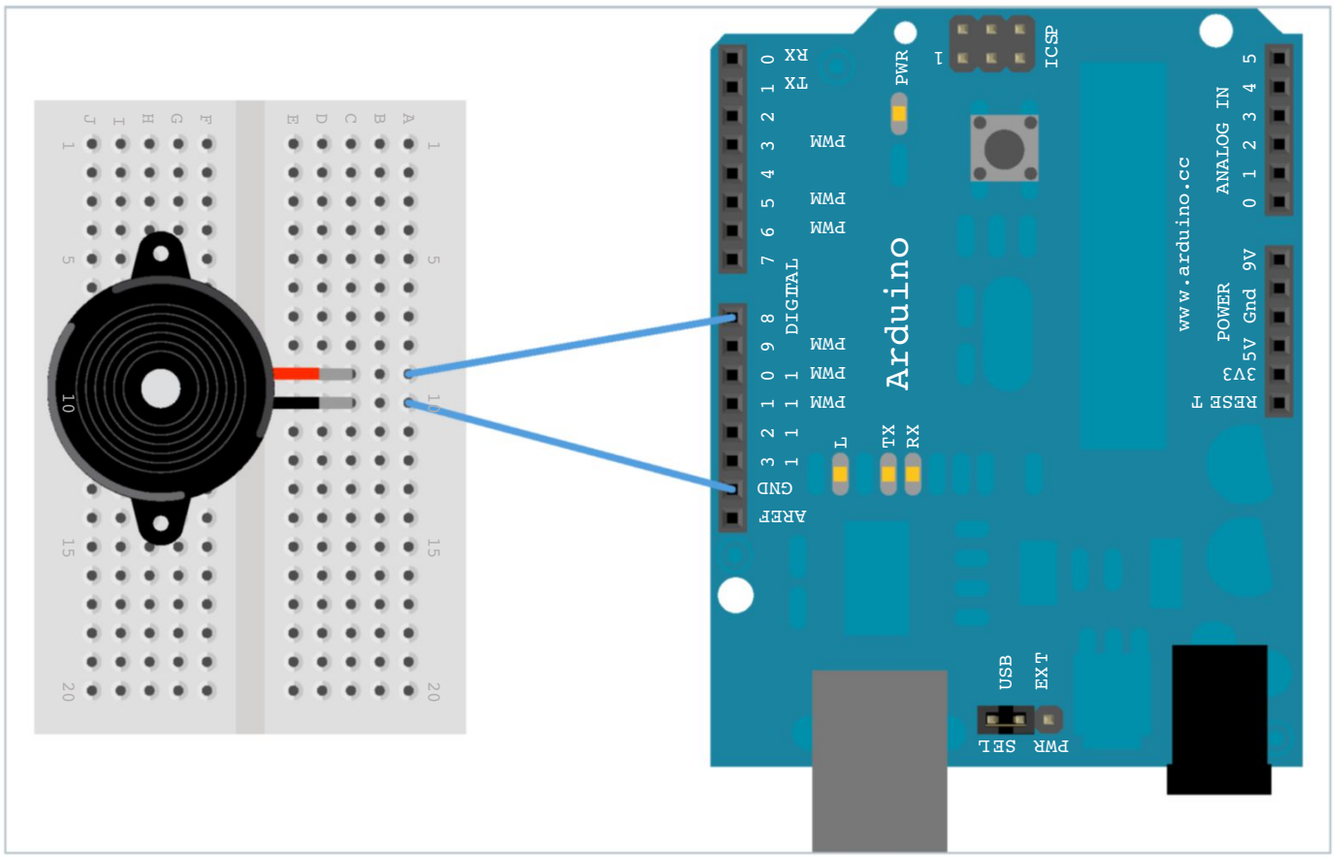


Fig 1 Connect the piezoelectric buzzer to Pin 8 The program will control the buzzer through Pin 8 on the Arduino

The music-box circuit

1. Connect the piezoelectric buzzer to Pin 8

As with the circuit we built last month, we're going to power the buzzer through the output Pin 8 on the Arduino, so wire the buzzer into that pin in order for the Sketch to control it. Ensure there's a solid connection – it's easy to miss the right slot on a breadboard (**Fig 1**).

2. Connect the piezoelectric buzzer to the Ground Pin

In order for electricity to flow, a complete circuit must be made. The second leg of the piezoelectric buzzer has to be connected to ground, but it doesn't matter which of the Arduino's ground pins you choose. Picking the one closest to Pin 8 makes the circuit neater (**Fig 2**, overleaf).

3. Check the connections

Because of the size of the body of the buzzer, which extends beyond its legs, it's easy to miss the right row if you're using a breadboard. Make sure that your wires lead from Pin 8 and Ground to the same row as the buzzer's legs, or the circuit won't work (**Fig 3**, overleaf).

> "In order for electricity to flow, a complete circuit must be made"

The interactive Sketch

Unlike our previous Sketch, which simply played its tune once, our new sketch will add an element of interactivity powered by a Perl-based back-end script, which we'll be coming to later on in the tutorial.

To construct this Sketch, we'll combine the toneMelody sketch from last month with the open source Alertuino Sketch developed by MCQN Limited – which also provides the Twitter-powered back-end script.

By far the easiest way to get this combined Sketch loaded is to snag a copy from this month's **Linux User & Developer** DVD. Make sure you've got all three files – twittermelody.pde, twittermelody.pl and pitches.h – in the same directory when you load the Sketch.

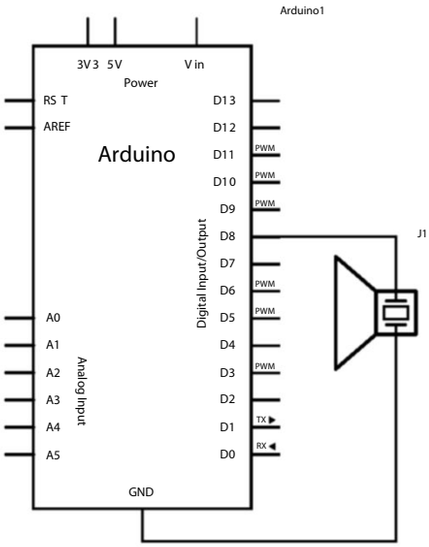


Fig 2 Connect the piezoelectric buzzer to the Ground Pin The buzzer must have a path to ground to complete the circuit

4. Extra variables for the Sketch

In addition to the variable declarations from our original Sketch, we've got some extra variables to think about this time around. The alertCount variable stores the number of mentions that our Twitter search has found, and controls the number of times the music plays. **The commandChar variable needs to be initialised at -1 to prevent the music from triggering early:**

```
int alertCount = 0;
int commandChar=-1;
```

5. Sketch setup

If you compare our modified Sketch to the one from last month, you'll notice that the setup section is significantly simpler. Because we want our melody to play on demand, the code for the tune has been moved to the loop section, which executes continuously while power is applied. **The setup section contains just two commands now: one to set the pin as an output, and the other to open a serial connection so that the Sketch can receive our instructions:**

```
void setup()
{
  pinMode(alertPin, OUTPUT);
  Serial.begin(9600);
}
```

6. The loop

Unlike our previous Sketch, which ran just once, we want our Twitter-powered music box to be ready at any moment for new mentions. **As a result, we're making far more use of the loop program section this time around – starting with an instruction to iterate through the number of alerts we've stored in alertCount:**

```
void loop()
{
  for (int i =0; i < (alertCount);
i++)
```

7. Serial reading

We'll skip over the melody-playing section of the Sketch, which won't execute yet and should be familiar from last month, and go straight to a new concept: reading an input from a serial connection. The Arduino is able to open a serial connection to a PC over the USB cable, and we want to watch this for messages from our Perl script. **We'll also output them to the serial console, for debugging purposes:**

```
 do
  {
   commandChar = Serial.read();
   if (commandChar != -1)
   {
   Serial.print(commandChar);
   }
  } while ( (commandChar < '0') ||
(commandChar > '9') );
```

> **We've got some extra variables to think about**

8. Counting the input

Because the input we're receiving from our Perl back end is a series of ASCII characters, we need to do some clever tricks to ensure that two- and three-digit numbers are counted correctly. **By taking the input a character at a time and multiplying our current values by ten, we can turn our string into the numerical value we need:**

```
if (commandChar != -1)
   {
   alertCount = alertCount * 10;
   alertCount = alertCount +
(commandChar - '0');
   }
```

9. The melody

When we've figured out how many times our Perl back end has seen our search term mentioned on Twitter, our Sketch loops back to the beginning and runs the melody code from last month one time for each mention. If you're happy with the default tune, you can hit the 'Upload' button and send the Sketch to your Arduino.

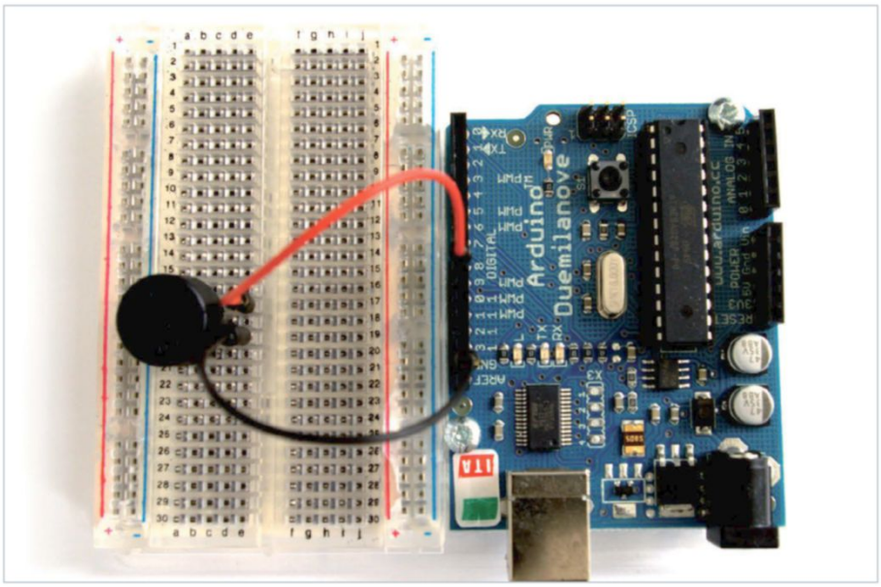


Fig 3 Check the connections Connect your wires to the same row as the buzzer's legs

The Perl back end

If you didn't hear any music when you uploaded the Sketch to the Arduino, something's gone wrong. Don't panic, because there's usually a simple explanation. Let's have a look at what you may have missed.

10. Prerequisites

Unlike our previous Sketch, this one isn't standalone: with no direct connection to the internet, the Arduino can't gather the information it needs about Twitter mentions. Instead, we're going to use a back-end program written in Perl to search Twitter, count the mentions for our chosen search time, and report that to the Arduino over the serial port.

The back end relies on a Perl module that isn't installed by default in most systems, which is used to parse the date from Twitter's RSS feed. To get the module in place, the easiest method is to install it via CPAN (**Fig 4**).

As the root user – or using sudo if you're on Ubuntu – install the RSS.pm module like so:

```
$ cpan DateTime::Format::RSS
```

11. Configure the serial connection

Open the file twittermelody.pl, which is the Perl-based back end that actually performs the Twitter search in your favourite text editor. The variable $alertuino_port should be set to the USB port that your Arduino is connected to – usually /dev/ttyUSB0. **If you're not sure, open the Tools>Serial Port menu in the Arduino IDE and make a note of the setting.**

```
$alertuino_port = "/dev/ttyUSB0";
```

12. Set the search term

The Perl back-end script searches Twitter for mention of a key phrase by accessing the site's search function via HTTP and counting the number of date stamps in the RSS file that it downloads. To customise the trigger phrase, change the $search_string variable. Remember that the search term will be part of an HTTP request, so a hash symbol should be encoded as %23 and a space as %20. In our example, we're searching for the hashtag #linuxuser.

```
$search_string = "%23linuxuser";
```

13. Activate the Arduino

Before you can run the script, you need to make sure that the Arduino is ready to receive serial communications. If you haven't uploaded the Sketch to the Arduino's microcontroller yet, do so now from the IDE with the Arduino connected via USB.

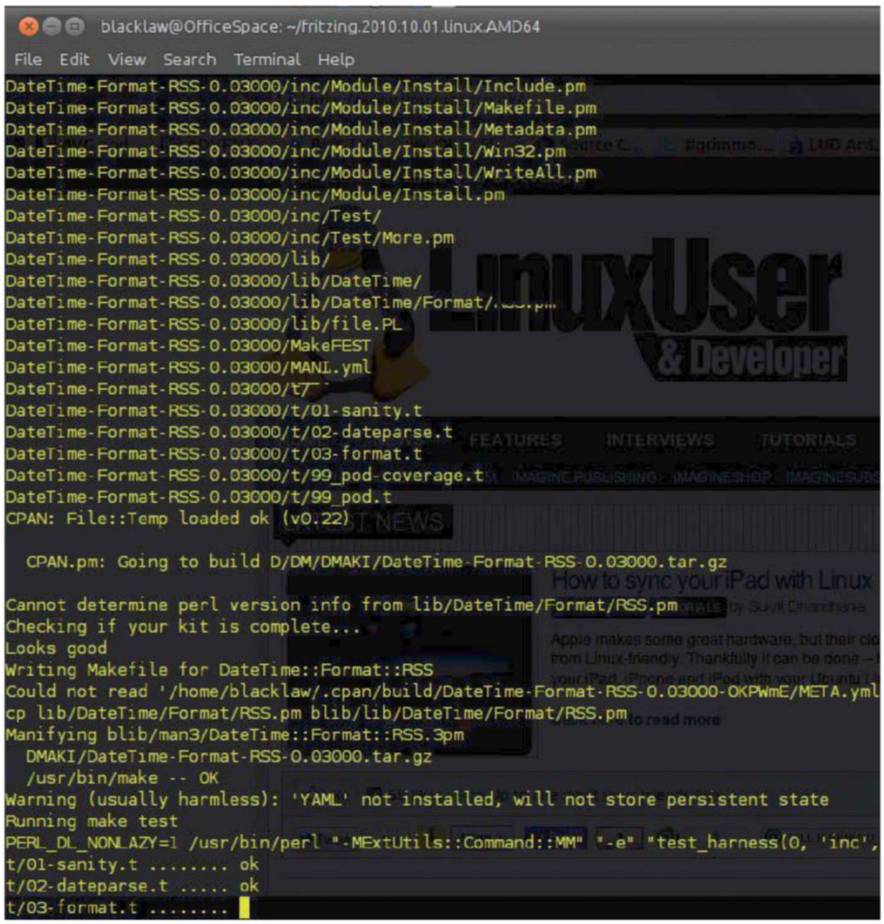


Fig 4 Prerequisites Install the date parser for Twitter's RSS feed via CPAN

When the receive light stops flashing on the Arduino and the upload is complete, press the Reset button. This makes sure that the serial connection is not held open by the IDE.

14. Run the back end

Once the Arduino is ready to receive serial communications, you need to active the Perl back-end script. **In a Terminal window, switch to the directory that contains the Perl file and execute it:**

```
$ perl twittermelody.pl
```

15. The result

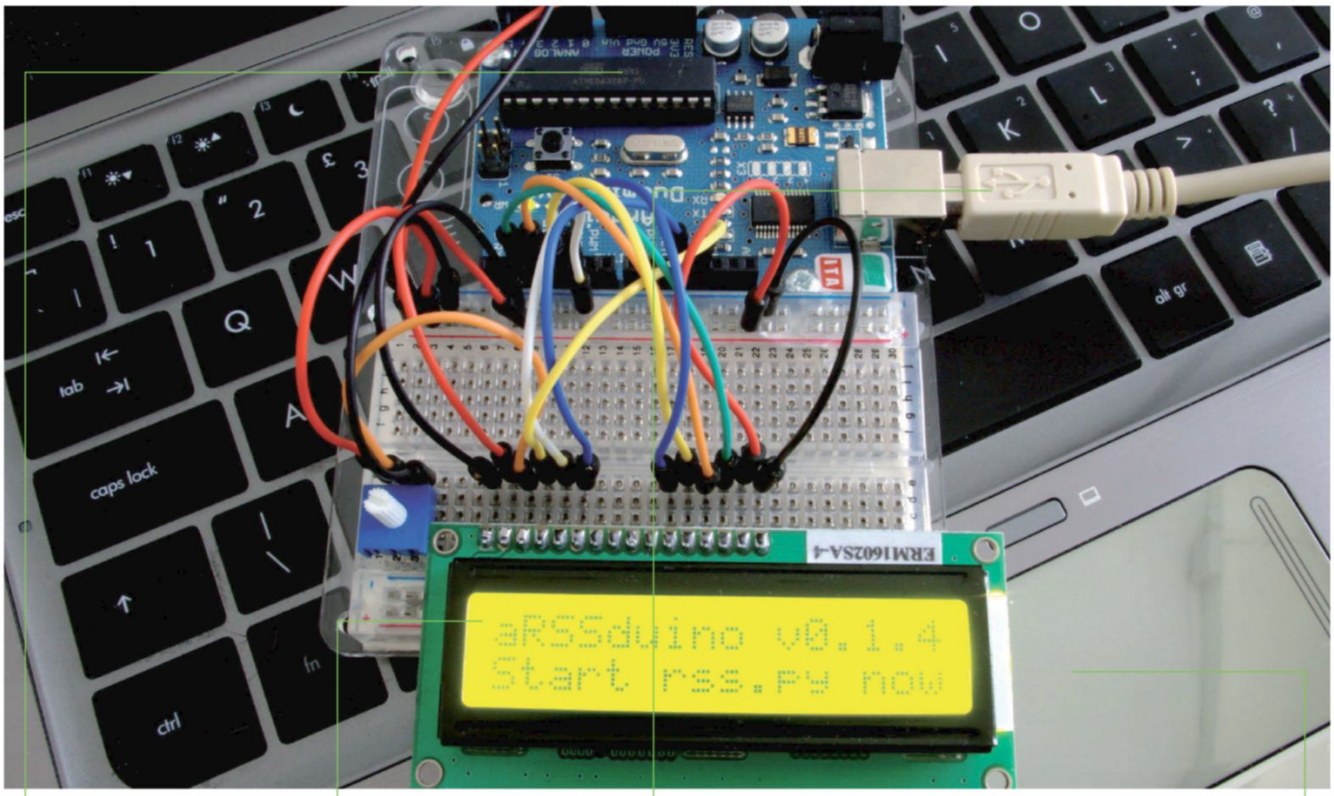
As the Perl back end downloads the Twitter search results as an RSS feed, it counts the number of new date fields since the last update – and, in the event of new results, triggers the Arduino to play a tune. For each new update found, the melody will be played once. Therefore if you've picked a popular search term for Twitter, you can expect to be hearing your chosen melody repeated quite a few times!

Conclusion

By teaming the Arduino's built-in serial communications system with a script running on our host PC, we can trigger events according to our whims – and not just by the results of inputs received directly by the Arduino.

While a Twitter-searching melody box isn't the most useful tool in the world, the same techniques could be employed to trigger a klaxon if system temperatures reach a critical level, or raise a white flag if a drive in a RAID system goes bad.

Next month, we'll be taking our project to the next level and adding an LCD panel to our Arduino circuit, giving us a secondary display for our back-end script to use.



The Arduino's microcontroller handles taking the input from the Python script and displaying it on the LCD

A liquid crystal display (LCD) provides a useful output for our last project, showing the results of Twitter searches

The USB connection provides both power and the serial interface for our back-end Python script

While this example is tethered to a laptop via USB, an Ethernet shield would allow you to make a standalone version

Developing with Arduino – part 3

Building on the last two projects, Gareth Halfacree adds a bit of complexity to the circuit and builds a secondary LCD display for Twitter search results for the final part of his Arduino tutorial series

Advisor

Gareth Halfacree has been breaking, fixing, tinkering and generally voiding the warranty on his electrical and electronic items for years, and hasn't received a fatal electric shock yet

If you've ever wanted to create your own electronic equipment, but never knew how, the Arduino project is for you. Originally developed as a teaching aid, it allows people with little or no electronics experience to build their own circuits and program the integral microcontroller with ease.

If you've been following this tutorial in previous issues, by now you'll be able to build a simple circuit with a piezoelectric buzzer and communicate with the Arduino via a serial connection. You will also be familiar with using the Arduino's serial-over-USB capabilities to take inputs from a script running on a host PC.

This month, we build on this knowledge by introducing a much more complicated circuit featuring a two-line, 16-character liquid crystal display (LCD), producing an ancillary display that we'll use to scroll through the results of a search operation on popular microblogging service Twitter.

Resources

16x2 Liquid crystal display with serial driver
http://oomlout.co.uk

Potentiometer

Arduino Experimenter's Starter Kit
http://oomlout.co.uk or

Arduino, breadboard and wiring

Arduino IDE
http://arduino.cc

The LCD circuit

The circuit to hook our LCD up to the Arduino's microcontroller isn't as complicated as it could be, thanks to the built-in serial controller on the LCD itself, but it's a bit more involved than our previous efforts. As before, take things slowly and double-check that your wires line up to the connections on the LCD – it's easy to miss which row of the breadboard you're wiring up.

1. Connect the LCD to the breadboard

If you've purchased a display with its own serial controller, it will be provided on a printed circuit board. Just push the pins into the breadboard towards the left-hand side, leaving room for the additional wiring and components you'll need on the right and three spare rows to the top (**Fig 1**). If you're using the ARDX kit from Oomlout, that puts the first pin of the LCD on row 7 on the mini-breadboard.

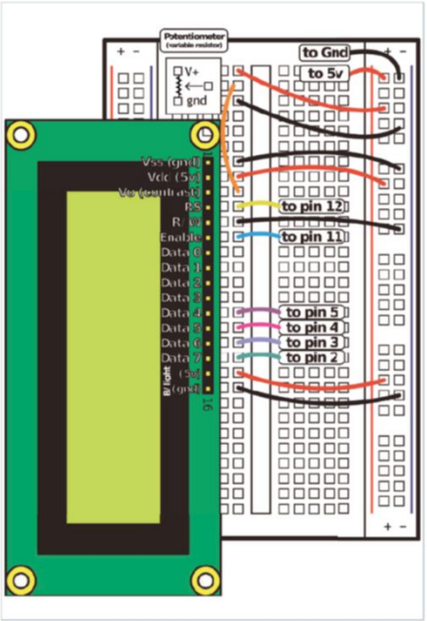


Fig 1 Connect the LCD to the breadboard Push the LCD's pins into the breadboard towards the left-hand side

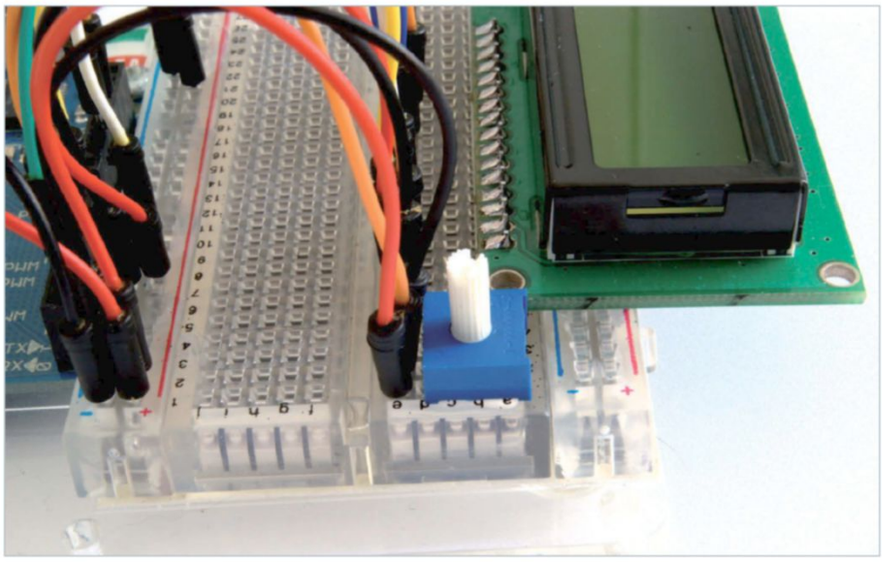


Fig 2 Connect the potentiometer It's that small white knob with the blue base

2. Connect the potentiometer

Our LCD allows the contrast of the display to be adjusted based on the input received by a certain pin. To adjust this, we'll be using a potentiometer. Connect the potentiometer, which looks like a small knob, to the spare three rows above the LCD – one pin per row (**Fig 2**).

> **"We need more than one source of power for our LCD. To obtain this, we'll tap into the power distribution rows of our breadboard"**

3. Power the breadboard

Unlike our previous circuits, we need more than one source of power for our LCD. To obtain this, we'll tap into the power distribution rows of our breadboard. Connect a wire from the +5V pin on the Arduino to the '+' column of the breadboard, and another from the GND pin to the '-' column.

4. Power the backlight

While this step is optional, if you don't provide the LCD's backlight pins with power it's hard to see the text in the dark. Connect the very bottom pin of the LCD, row 22 of the breadboard if you're using the Oomlout ARDX, to the '-' column of the breadboard – and the next pin up, row 21, to the '+' column. This provides the +5V needed to turn on the backlight.

5. Connect the data pins

To make our text appear on the LCD, we need to connect the Arduino's microcontroller to its data pins. Although the LCD has eight data pins, for our purposes we only need to connect the bottom four on rows 20 to 17. Connect row 20 to pin 2 of the Arduino, row 19 to pin 3, row 18 to pin 4, and row 17 to pin 5.

6. Connect the reset and enable pins

Before the LCD is ready to receive input from our data pins, it must be enabled. To do so, connect the Enable pin on row 12 to pin 11 on the Arduino. We'll also want to hook up the Reset pin on row 10 to pin 12 on the Arduino (**Fig 3a** & **3b**).

7. Connect the LCD's power

Before the LCD can work, it needs power. Connect the pin on row 7 to the '-' column of the breadboard, and the pin on row 8 to the '+' column of the breadboard. This provides the +5V that the LCD requires.

8. Wire up the potentiometer

To use our potentiometer to control the contrast of the LCD, we need to provide it with power and connect it to the LCD's Contrast pin. Connect row 1 to the '+' column and row 3 to the '-' column for power, then wire the centre pin on row 2 to the Contrast pin on row 9.

When our LCD is powered on, twisting the potentiometer will adjust the contrast. Too far one way, you'll see only black blocks; too far the other way, you'll see nothing at all.

9. Connect the R/W pin

The LCD has a special pin marked R/W, which must be connected to ground before the screen's controller will accept commands. Wire the pin on row 11 to the '-' column at the side of the breadboard, then the circuit will be complete.

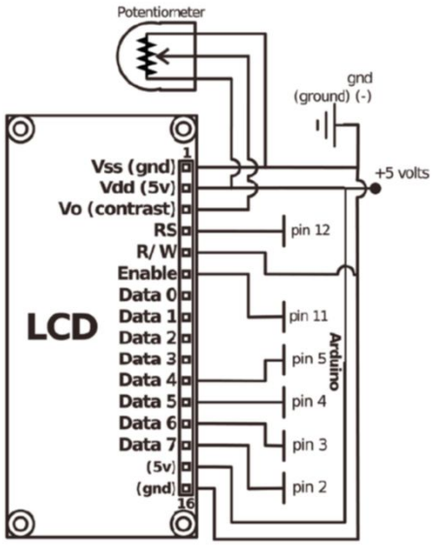


Fig 3a Connect the reset and enable pins A diagram of the connections needed

The Python back-end

Unlike our previous project, which used a script written in Perl, we're going to use a Python back-end to search Twitter and feed the results to the Arduino. When communicating over serial, it doesn't matter what language you write the back-end in – feel free to use whatever you're most comfortable with.

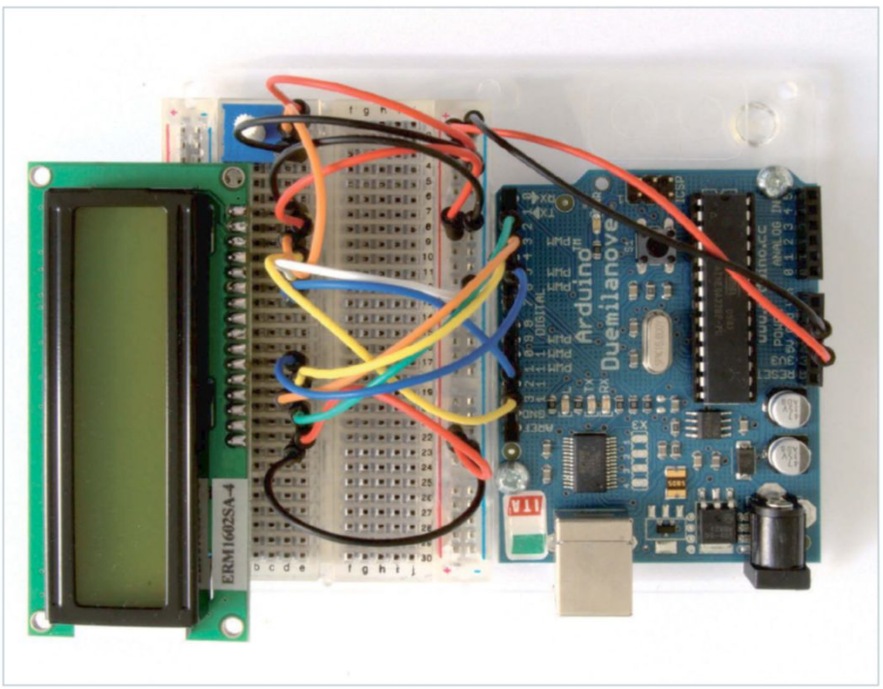


Fig 3b Connect the reset and enable pins You need to do this to enable the LCD

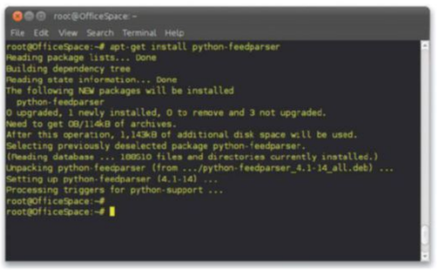


Fig 4 Install FeedParser Python library

1. Install FeedParser

We'll be accessing our Twitter search results via an RSS feed. And to make it easy to pull the details we need out of the stream, we'll be using the Universal Feed Parser Python library by Mark Pilgrim (**Fig 4**).

To install the library, use your package manager as root to download and install the python-feedparser package:

```
$ apt-get install python-feedparser
```

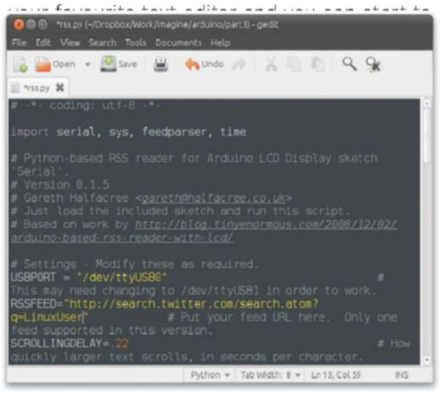
or

```
$ yum install python-feedparser
```

If the Universal Feed Parser isn't available in your distribution's repositories, you can download it from **www.feedparser.org**. If you don't install the library, you'll get an error when you run the Python script.

2. The Python script

The grunt-work – searching Twitter, downloading the RSS feed and formatting its content for the LCD – is carried out by a Python script called rss.py, which you'll find on the cover DVD this issue. Load this script into



■ The Python script handles the nitty-gritty

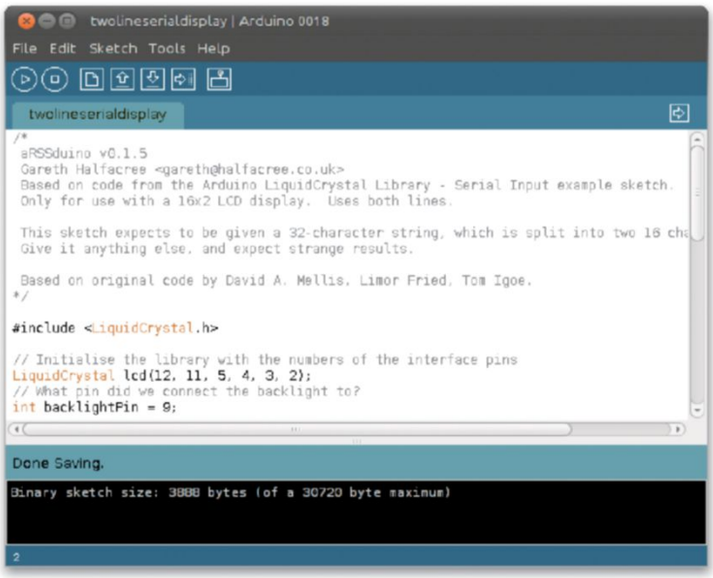


Fig 5 Load the Sketch It's on the cover DVD

customise it to your needs.

3. Changing the search term

While you can customise much of the script – including the speed at which it scrolls – the main thing to look at is the RSSFEED variable. The first part tells the script to visit Twitter's search page and retrieve an RSS feed, while the second is the term you search for. If you want to search for a space, remember to encode it in the URL as %20.

```
RSSFEED="http://search.twitter.com/search.atom?q=LinuxUser"
```

The LCD display Sketch

The Sketch we're using on the Arduino is a modified version of the default SerialDisplay example found in the LiquidCrystal library of the Arduino IDE. Unlike the original, however, it allows us to display on both lines of the LCD – and scroll just the bottom line – thanks to a bit of cheating.

1. Load the Sketch

The Sketch is called twolineserialdisplay.pde, and can be found with its Python back-end script on the cover DVD. Load it in the IDE from the File – Open menu, or copy it to your SketchBook (**Fig 5**).

2. Writing to the LCD

The LiquidCrystal library, included as standard with the Arduino IDE, allows simple writing to the LCD. Using the preface 'lcd', the commands 'clear' and 'print' do what you might expect, while setCursor allows you to place the cursor anywhere onto the display – including the second line.

```
lcd.clear();
lcd.print("aRSSduino v0.1.5");
```

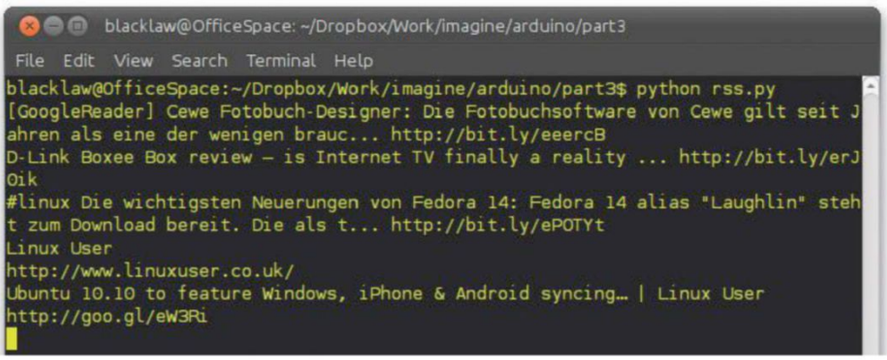


Fig 6 Run the Python back-end The Terminal displays the content of each result

```
lcd.setCursor(0,1);
lcd.print("Start rss.py now");
```

3. Talking to the LCD

While those commands are fine for writing from our Sketch to the LCD, we want to take input from the serial port. To do that, we can chain the lcd.write command with Serial.read.

To get only one line of the two-line display to scroll can be challenging in the Arduino IDE, so we're going to cheat and accept a 32-character input from our Python back-end and display each block of 16 characters on its own line. The Python script will then handle scrolling the bottom line.

```
lcd.setCursor(0,0);
  while (Serial.available() > 16)
{
  lcd.write(Serial.read());}
  lcd.setCursor(0,1);
  while (Serial.available() > 0) {
  lcd.write(Serial.read());
  }
```

4. Upload the Sketch

Plug the Arduino in to the PC using the USB cable and hit the 'Upload' button to program the microcontroller. When you connect the Arduino, the LCD should light up – and when the upload is complete, a message will appear on the display.

5. Run the Python back-end

Keeping the Arduino connected to the USB port, drop to a terminal and change to the directory with the rss.py file in it. **Run this file in Python:**

```
$ python rss.py
```

The Terminal will display the content of each result in the Twitter search (**Fig 6**), while the LCD should show the user's Twitter name on the top line and their message below.

If you've got this far, then you've got a good idea of the sort of things that can be achieved with an Arduino prototyping kit – and there's plenty of scope for expanding on the techniques we've covered here.

Adding 'shields' to your design allows access to additional hardware, such as the Ethernet shield for creating standalone hardware that can connect to the internet without the need for a host PC. Other shields add wireless connectivity, LED multiplexers, or even touch-screen displays for more advanced projects.

To expand your Arduino knowledge, try the Arduino Playground at **http://www.arduino.cc/playground/**, or the collection of projects at electronics specialist Oomlout's site at **http://www.oomlout.co.uk**.

Multi-boot made easy

Kunal Deo shows us how to build a multi-boot system capable of booting Windows, multiple Linux distros and BSD

Unless you are building your own PCs or you have managed to convince the seller not to install Windows (and save few bucks), most of the PCs and computers we buy today come pre-installed with Microsoft's operating system.

Before we show you how to escape this sorry state, you might be confounded by a number of ridiculous myths surrounding the removal or changing of the Windows install on your shop-bought system. The most prominent one is that installing Linux on a system pre-installed with Windows will void the warranty. Yes, it's a myth. Another popular one is that installing Linux on your system to coexist with Windows involves practising some sort of dark magic requiring a premium call-out by Count Dracula to get it done. There are other outrageous myths out there, but the point is that making your system dual-boot or multi-boot with Linux or any other operating system is not a dark art and will not void your system's warranty.

Before we get down to the business of installing and configuring multiple operating systems in a multi-boot configuration, let's take a quick look at what you'll need to have ready...

A system with Windows 7 (or XP) installed as the default operating system. If you do not have Windows installed, even better: you can skip the 'Preparing the hard drive' step and use a GParted disc (as mentioned in that step) to resize your existing Linux partitions. Please keep in mind that partition numbers will change accordingly, though.

Hard drive with a single partition formatted with NTFS (and maybe a Recovery partition) which also holds the Windows OS. Again, it is not mandatory, but was assumed for the sake of simplicity. If you have more than one NTFS partition, shrink the partition with the most free disk space and continue accordingly. Please keep in mind that in this case partition numbers will change.

DVD drive

Ubuntu Linux 10.10 http://www.ubuntu.com/desktop/get-ubuntu/download

openSUSE 11.4 http://software.opensuse.org/114/en

Mandriva Linux 2010.2 http://www.mandriva.com/en/downloads/free/

PC-BSD 8.2 http://www.pcBSD.org/content/view/202/11/

Configuring a multi-boot system

It is time to jump into the multi-boot action. Before you leap in, though, make sure you have a safe fallback plan in case things go wrong. It is recommended that you back up your data before configuring a dual or multi-boot system; in fact, it is important to have data backup before you make any significant changes to your system

The plan

As any good civil engineer will tell you, planning is key to building a successful skyscraper. Our plan is to build a multi-boot system, which is a heterogeneous combination of Windows, Linux and other UNIXes (such as PC-BSD).

For this tutorial, we have decided that our system will contain the following operating systems...

1 Windows 7 or XP (pre-installed)
This OS should already be installed on your system. Maybe you installed it or came pre-installed with the system you bought.

2 Ubuntu Linux 10.10
This will be the second operating system for our multi-boot setup.

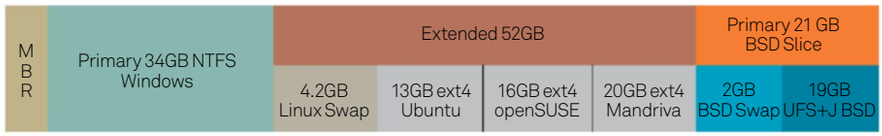
3 openSUSE Linux 11.4
This will be our third operating system.

4 Mandriva Linux 2010.2
This will be the fourth operating system for our multi-boot setup.

5 PC-BSD 8.2
PC-BSD is a UNIX-like, desktop-oriented operating system based on FreeBSD. This will be the fifth operating system for our multi-boot setup.

We mentioned the version number for obvious reasons, but you can try out other versions as well. They will work the same, as long as the installation wizard is not significantly different in the version you are installing

Our multi-boot configuration has 1x Windows OS, 3x Linux distributions and 1x FreeBSD operating system. Our setup is based on an assumption that you have a single 100GB partition on which Windows is already installed. You can easily adopt the whole process to your own configuration as well.



Partition plan

We will be partitioning our 100GB hard drive as shown in the diagram above.

Primary 34GB NTFS – /dev/sda1: We have not created this partition, but it was already present on the hard drive as a result of Windows installation. We have shrunk it to 34GB to make room for other partitions. This partition holds Windows 7 or Windows XP.

Extended 52GB – /dev/sda2: An MBR partitioned disk has a limit of carrying only four primary partitions, but we need more than that. So we have created an extended partition of size 52GB which further acts as a container for several logical partitions that will be used for our Linux installations. This partition will contain the following four logical partitions...

Logical 4.2GB Linux Swap – /dev/sda5: This partition will hold the swap partitions which will be shared across all installed Linux distributions. Did you notice the odd numbering with /dev/sda5? The way it works is that 1 to 4 are reserved for four primary partitions, hence we're using 5 to 8 for secondary ones. Note that we can use the swap partition to share with other operating systems such as FreeBSD. Although there are ways around that, we'll avoid them for the sake of simplicity.

Logical 13GB ext4 Ubuntu – /dev/sda6: This partition will be used to install Ubuntu Linux. You can use other file systems as well, but we have decided upon ext4 as it is the default for many popular Linux distributions.

Logical 16GB ext4 openSUSE – /dev/sda7: This partition will be used to install openSUSE Linux.

Logical 13GB ext4 Mandriva – /dev/sda8: This partition will be used to install Mandriva Linux.

Primary 21GB FreeBSD Slice – /dev/sda3: In the FreeBSD world, partitions are called slices. These slices can also hold other partitions, just like the extended partitions. In FreeBSD terms, this slice is numbered as da0s3. This slice holds the following two partitions...

BSD Swap Partition

UFS+J Formatted Root Partition: This partition will be used for the PC-BSD installation.

Boot plan

Now that we have planned how we are going to partition the disk, let's plan out the boot system. This is very important because if you cannot get this plan right, you will not be able to boot (or in other words start) the operating systems. The plan is to use Ubuntu as the primary boot loader for all the installed OSs. When we say Ubuntu we mean Ubuntu's bootloader, ie GRUB 2, which will be used to boot all the installed OSs. For this reason we will be installing Ubuntu's bootloader onto the MBR (master boot record), which is the default installation setting for Ubuntu's installer. MBR is loaded directly by the system BIOS, hence it will always load Ubuntu's bootloader when the computer is started.

Loading other operating systems through Ubuntu's bootloader will be achieved by a process called chainloading. As the name implies, in this process the bootloader passes the control of the boot sequence to another bootloader, located on the device to which the menu entry points. This can be a Windows operating system, but also any other, including Linux. During the installation process, Ubuntu already has an entry for Windows using the same method.

By default, all Linux distribution installers try to install their respective bootloader on the MBR, which is bad as we do not want to overwrite Ubuntu's bootloader. We will customise the installation process to make sure that other distributions install their respective bootloaders in their respective root partitions (ie the same partitions in which they are being installed). Then we will use chainloading to load the respective bootloaders to boot into the respective operating systems.

Now that we have done our planning, let us jump right into the action.

Preparing the hard drive

Before we go onto installing, we need to prepare the hard drive for multi-boot installation. This includes making sure we have the necessary free space to create partitions for other operating systems. **In the following steps we will use the Windows Disk Manager utility to shrink the existing partitions to make way for new ones.**

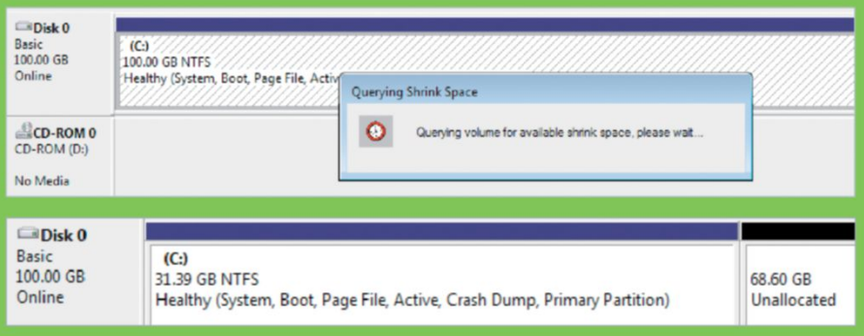
1. Boot into Windows 7 (or Windows XP).

2. Open the Disk Manager utility. You can open it by hitting Start, then typing diskmgmt.msc in the search box and pressing Enter.

3. Right-click on the C drive and click Shrink. This process may take some time.

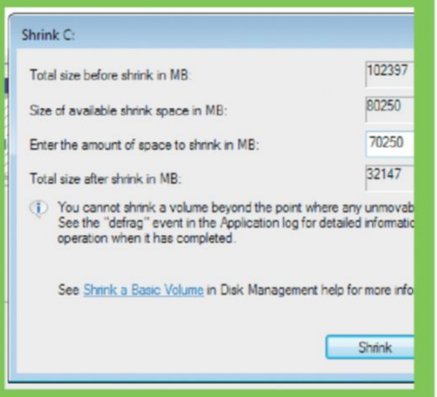
4. On the next screen, enter the amount of space in MB (1,024MB = 1GB) to shrink by, from the available free space on this volume, for the new partition and click Shrink. Make sure that you shrink the space enough to install all the operating systems you will be installing. For our tutorial, 60 to 80GB of unallocated space should be sufficient.

5. After a few minutes you will have an unallocated space left. We will use this space to create partitions for the FreeBSD installation.



■ Shrinking the Windows partition

Note: The size of the available space can be restricted by the amount of space currently allocated on the hard drive for the virtual memory page file, System Restore max size and hibernation files. If the size provided by the Disk Manager is inadequate, retry after turning off these features.



■ The value in the 'Total size before shrink' box indicates the current size of the partition; the value in the 'Size of available shrink space' indicates the maximum size that you can allocate to the new second partition

Partitioning the hard drive

All Linux distributions (for that matter, all operating system installation programs) offer hard drive partitioning during their install process; however, it is always helpful to partition your hard drive before installing the operating system. This helps in clear planning and safe execution of multi-OS installation.

Note

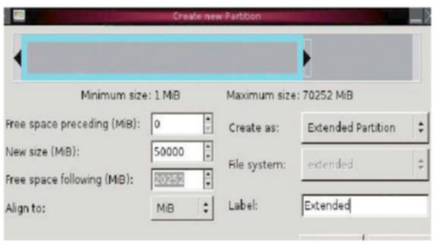
You will need to configure your system to boot from the optical media drive. If your system has a boot menu (which allows you to choose a drive to boot from), you do not need to configure it in the BIOS – just use the key to directly boot into the boot menu.

1. Download and burn GParted live ISO from http://gparted.sourceforge.net/download.php.

2. Boot from the GParted live CD. While booting the GParted live CD, select the default options. Proceed after GParted has started up.

3. Right-click on Unallocated and select New. The 'Create New Partition' window will open.

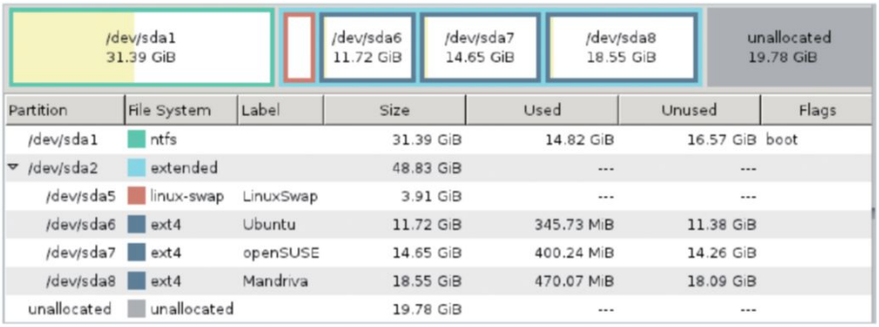
4. Select 'Create as:' Extended Partition. Set the New Size around 50,000MB (sufficient to accommodate three Linux distros). Leave the rest of the values at default. Click Add. You have just added a single extended partition which will hold our logical partitions for OS installations. In the next step we will create logical partitions.



■ Creating the extended partition

5. Linux swap partition: A swap partition acts as virtual memory; it is used when the system is low on physical available memory (RAM). In this step we will create a swap partition which will be shared across all the installed Linux distributions. As a general rule of thumb, a swap partition should be equal to twice the physical RAM installed in the system. Right-click on the newly created extended partition and set the values as:

New Size: 4,000MB
Create as: Logical Partition
File system: linux-swap



Partition	File System	Label	Size	Used	Unused	Flags
/dev/sda1	ntfs		31.39 GiB	14.82 GiB	16.57 GiB	boot
▾ /dev/sda2	extended		48.83 GiB	---	---	
/dev/sda5	linux-swap	LinuxSwap	3.91 GiB	---	---	
/dev/sda6	ext4	Ubuntu	11.72 GiB	345.73 MiB	11.38 GiB	
/dev/sda7	ext4	openSUSE	14.65 GiB	400.24 MiB	14.26 GiB	
/dev/sda8	ext4	Mandriva	18.55 GiB	470.07 MiB	18.09 GiB	
unallocated	unallocated		19.78 GiB	---	---	

Fig 1 Final disk layout

Glossary

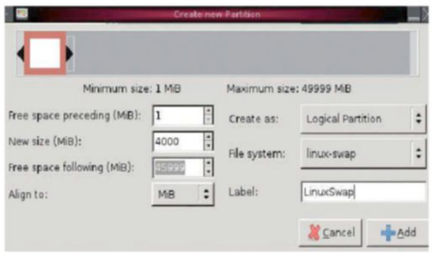
Explanations for a few of the terms used in this tutorial

MBR: All disks start with a boot sector. When you start the computer, the code in the MBR (master boot record) executes before the operating system is started. The location of the MBR is always track (cylinder) 0, side (head) 0, and sector 1. The MBR contains a file system identifier.

Ext4: The ext4, or fourth extended file system, is a journaling file system developed as the successor to ext3. Ext4 is the default file system for most Linux distributions.

Bootloader: A bootloader is the first software program that runs when a computer starts. It is responsible for loading and transferring control to the operating system kernel software. The kernel, in turn, initialises the rest of the operating system (eg Linux).

GRUB 2: GNU GRUB is a multi-boot bootloader. It was derived from GRUB, the GRand Unified Bootloader, which was originally designed and implemented by Erich Stefan Boleyn.



Creating the Linux swap partition

6. Logical ext4 partitions: These partitions will hold the root file system (/) of the respective Linux distributions. Right-click the unallocated space in the extended partition and select New, then set the values as follows:

New Size: 12,000MB
Create as: Logical Partition
File system: ext4

Repeat this step two more times. You can set the size of partition according to your needs. We recommend allocating at least 10,000MB for each Linux distribution.

7. Applying changes: Verify the partition map to see if everything is OK. Then click Apply to write the changes to the disk. The final disk layout should look similar to **Fig 1**.

Installing the operating systems

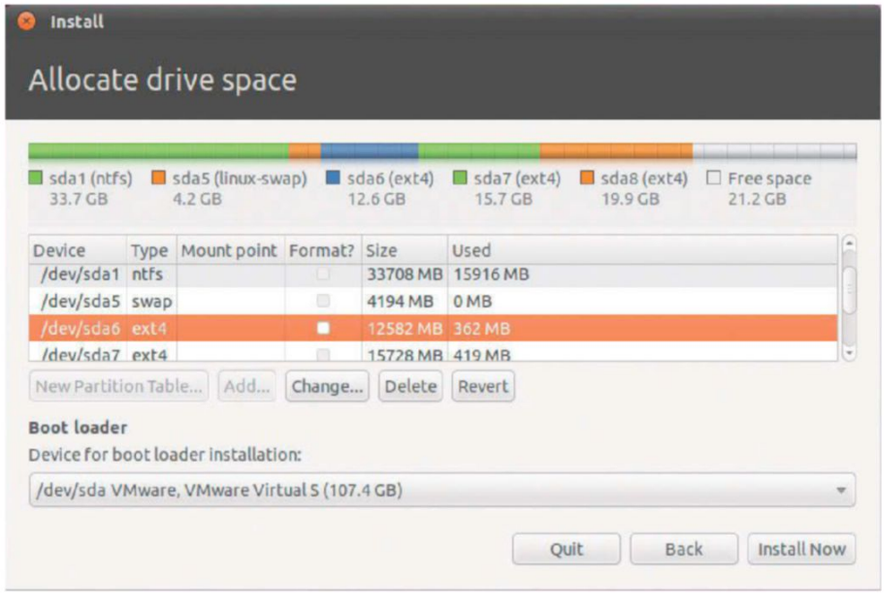


Fig 2 Ubuntu Partition Manager

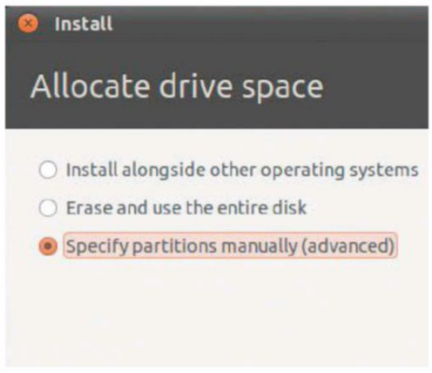
Ubuntu 10.10

Perform the following steps to install the second operating system

01 Insert the Ubuntu installation disc and boot from it.

02 Continue using the installation wizard until you reach the 'Allocate drive space' step (**Fig 2**).

03 Select 'Specify partitions manually (advanced)' and click Forward.

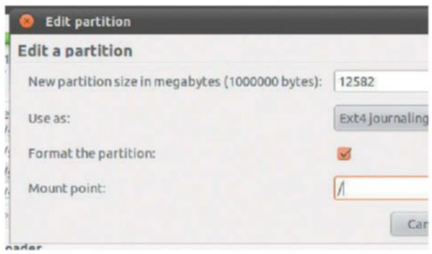


Allocate drive space

04 Select the first ext4 partition and click Change to open the Edit Partition dialog.

05 Select the options as follows, and then click OK...

Use as: Ext4 journaling file system (or any other partition you prefer)
Format the partition: Selected
Mount Point: /



Editing Partition Settings

06 Then click Install Now to continue with the installation.

07 Finish the installation wizard as usual and then restart your system. At this point you should receive a boot prompt upon restarting your system. Ignore this for now and proceed with installing the next Linux operating system: openSUSE.

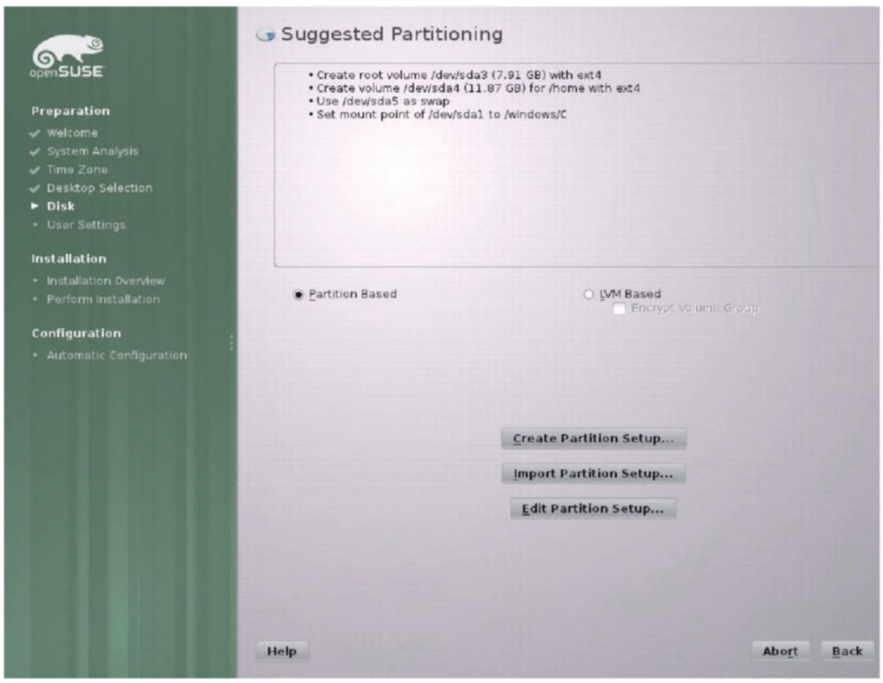


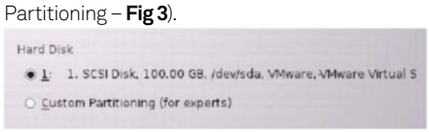
Fig 3 openSUSE Suggested Partitioning

openSUSE Linux

Perform the following steps to install the third operating system

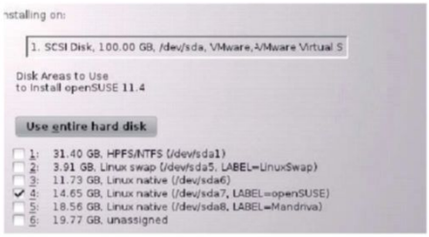
01 Insert the openSUSE Installation Disc and boot from it.

02 Continue the installation wizard till you reach the Hard Disk step (or Suggested Partitioning – **Fig 3**).



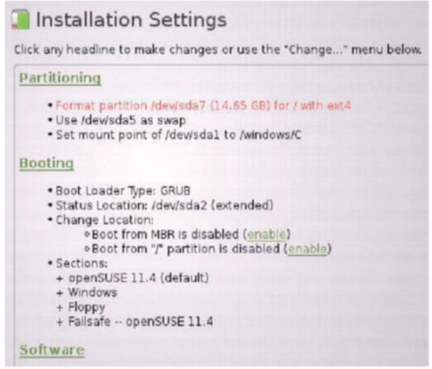
Hard disk selection

03 Since we already have our own plan for installing openSUSE, we will ignore the suggested partitioning – instead, click Create Partition Setup... You will then be presented with Hard Disk list. Select the first hard disk and click Accept.



04 On the next screen you will be presented with the list of partitions available on the selected disk. Select 'Partition for openSUSE installation'. It should should show 'LABEL=openSUSE' and should be fourth in the serial order. Do not select any other partition check. Under proposal type, uncheck everything as we do not want a separate home partition. Click Accept to continue.

Warning: Please be advised that if you select a partition which already contains data or an operating system installation, you may lose the data on the respective partition.



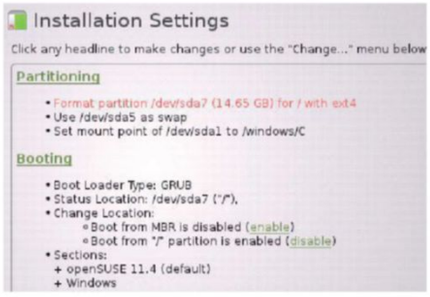
05 Continue the installation wizard till you reach the Installation Settings screen. Here we will configure the Boot setting for openSUSE. Click on Booting (green colour text) to open Boot Loader Settings.

Boot Loader Settings page

06 In the Boot Loader Settings page, click on the tab Boot Loader Installation. Under the Boot Loader Location, just check 'Boot from Root Partition'. Uncheck all other options. This will ensure that the openSUSE bootloader only gets installed on the root partition and not on the master boot record, as we do not want to overwrite the Ubuntu bootloader (which is installed on the master boot record).

07 Click on Boot Loader Options on the Boot Loader Setting page to open the Boot Loader Options page. Here, under Boot Menu, uncheck 'Write generic Boot Code to MBR'. Leave the rest of the options as default. Click OK to apply the changes and return to the Boot Loader Settings page.

Warning: Failing to do this step will result in a non-bootable system.



08 Click OK on the Boot Loader Settings page to return to the Installation Settings page. Here you will see a summary of the configuration that you have just created.

Once you are satisfied with the changes you have made, click Install to continue with the rest of the installation process. Note that you will not be able to boot openSUSE immediately after installation. We will configure it later on in this tutorial.

Fig 4 Partitioning step

Mandriva Linux

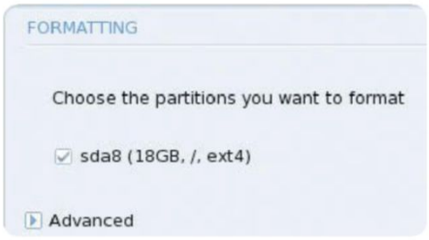
Perform the following steps to install the fourth operating system

01 Insert the Mandriva Linux installation disc and boot from it.

02 Continue the installation wizard till you reach the Partitioning step (**Fig 4**). Here select 'Use existing partitions' and click Next.

Mounting the root partition

03 On the next screen of Partitioning you will be asked to select the mount points. By now, the first two partitions should already be occupied by Ubuntu and openSUSE Linux. Type '/' in the last partition (which is still empty) and clear out the remanning partitions. Click Next to continue.



Formatting the root partition

04 On the next screen you will be presented with the Formatting option. Leave the default option selected and click Next. This starts formatting that partition and begins the installation process.

05 Continue the installation steps till you reach the summary page (**Fig 5**).

06 Click on the Configure button, present after the text 'Bootloader – grub-graphic on /dev/sda', to open the Boot Loader Main Options page (**Fig 6**). Here, set the 'Boot device' as your root device. The root device is the same partition which you mounted as '/' in step 3. Click Next to continue.

"By now, the first two partitions should already be occupied by Ubuntu and openSUSE Linux"

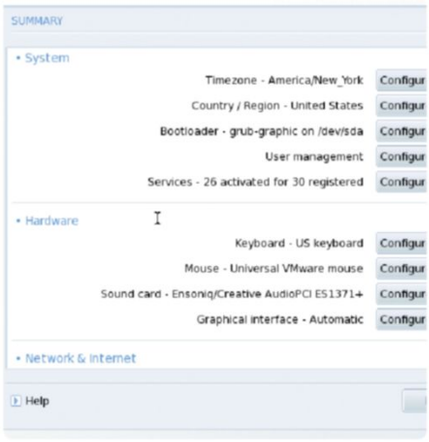


Fig 5 Installation Summary Page (Before Configuration)

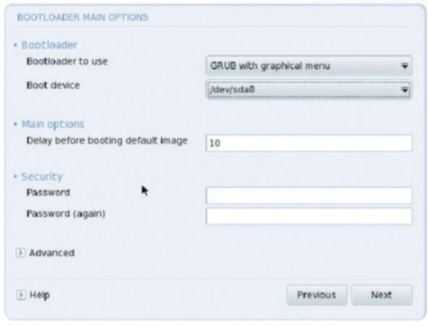


Fig 6 Boot Loader Main Options

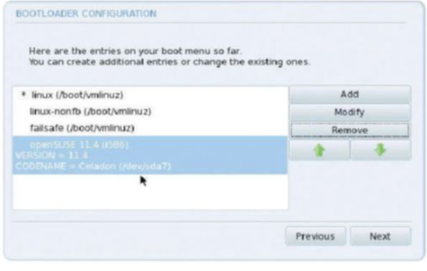


Fig 7 Boot Entries

07 Remove any entry related to other Linux distributions, in this case openSUSE 11.4 . You can remove it by selecting the entry then clicking Remove (**Fig 7**). Click Next to continue. You can now proceed with the rest of the installation as usual.

Recovering the Windows bootloader

There may be times when you want your Windows bootloader to return to its old place, the MBR. Maybe you want to uninstall Linux distributions from your system or you have accidentally deleted the Linux partition which also holds the files for the bootloader currently installed on the MBR, hence rendering the system unbootable.

To recover the Windows XP bootloader:

1. Boot from the Windows XP installation media.

2. At the installation prompt, choose R to repair an existing installation.

3. Select '1) C:\Windows', then press Enter. If you have set an admin password, enter the password.

4. At the prompt, type in the following commands in order...

C:\> FIXBOOT C:
C:\> FIXMBR
C:\> BOOTCFG /rebuild

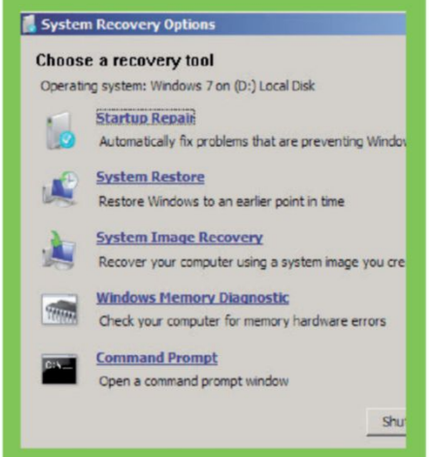
To recover the Windows 7 bootloader:

1. Boot from the Windows 7 installation media. If you don't have the Windows 7 installation media, you can download a trial from Microsoft Technet: http://technet.microsoft.com/en-gb/evalcenter/cc442495.aspx?ITPID=wtcfeed.

2. Click 'Repair Your Computer' at the prompt.

3. Select Windows 7 and click Next.

4. Click 'Startup Repair' to reinstall the Windows 7 bootloader.



■ Windows 7 Startup Repair

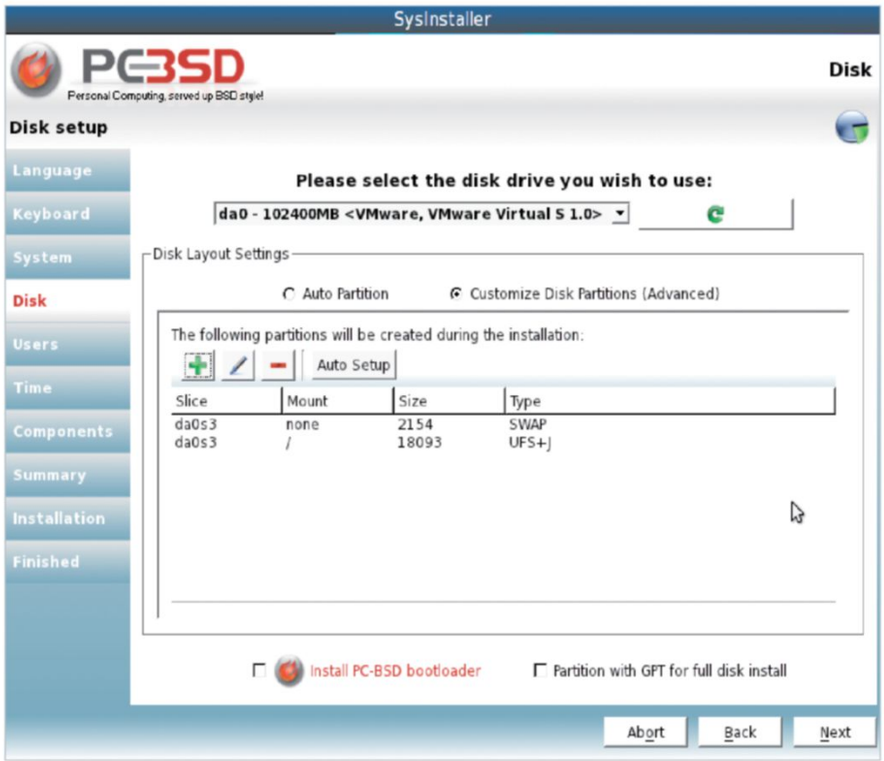


Fig 10 Disk Layout Settings

PC-BSD

Perform the following steps to install the fifth operating system

01 Boot from the PC-BSD installation disc and continue till you reach the Disk Layout Settings step.

02 Select 'Customize Disk Partitions (Advanced)' and click the '+' sign. In the File System window (**Fig 8**), select Unused Space from the Disk / Slice drop-down menu. Select SWAP from the Type drop-down, set the size as 2000MB and click Save. This will create the Swap partition for PC-BSD.

03 Click the '+' sign again. In the File System window select Unused Space from the Disk / Slice drop down menu. Select UFS+J (you can also select other FS type of your choice). Set the mount point as '/'. Slide the Size slider to the right (maximum) and click Save (**Fig 9**).

04 At the Disk Layout Settings screen (**Fig 10**), make sure that 'Install PC-BSD bootloader' and 'Partition with GPT disk for full install' are both unchecked. Click Next to continue with rest of the installation process.

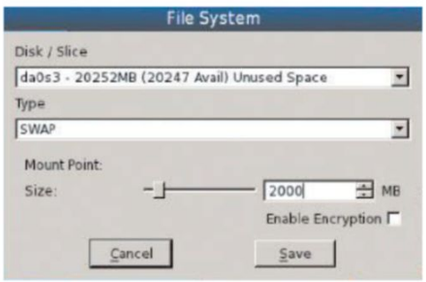


Fig 8 Creating swap partition for PC-BSD

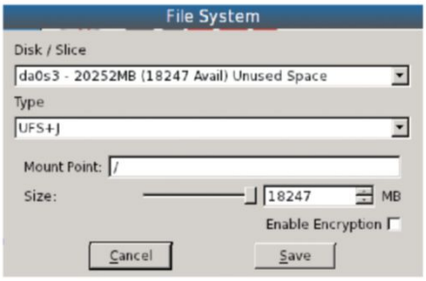
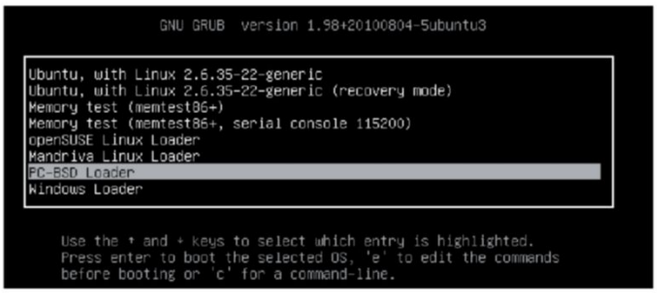
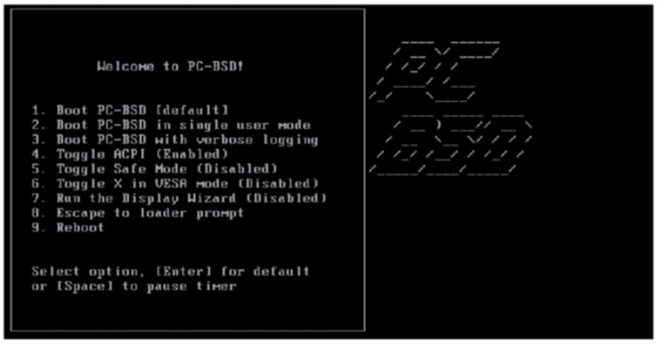


Fig 9 Creating root partition for PC-BSD





Configuring the Ubuntu bootloader (GRUB 2)

As mentioned in the planning section, we will be using Ubuntu's bootloader to boot all other operating systems

Start Ubuntu Linux and perform the following steps to configure the bootloader...

1. Stopping automatic boot entry addition: By default, Ubuntu's bootloader (GRUB 2) is configured to automatically discover and add the Linux kernel to the boot entry. While this may sound fairly nice, it actually adds unbootable boot menus because it detects the root FS for other operating systems in an incorrect way. Hence we need to stop automatic boot entry addition:

```
$ sudo chmod -x /etc/grub.d/30_os-prober
```

This command removes the execute privileges from os_prober which are responsible for detecting other operating systems.

2. Adding chainloader entries to installed operating systems: Open the file /etc/grub.d/40_custom (with '$ sudo gedit /etc/grub.d/40_custom') and add the following lines:

Chainloader entries

```
menuentry "openSUSE Linux Loader" {
set root=(hd0,7)
chainloader +1
}

menuentry "Mandriva Linux Loader" {
set root=(hd0,8)
chainloader +1
}

menuentry "PC-BSD Loader" {
set root=(hd0,3)
chainloader +1
}

menuentry "Windows Loader" {
set root=(hd0,1)
chainloader +1
}
```

Make sure that the root entries are relevant to your system.

3. Applying changes: Enter the following command to update the GRUB configuration:

```
$ sudo update-grub
```

That's it, we have achieved the multi-boot configuration. Reboot your system and start playing in the multidimensional world of different operating systems. As you have may noticed, it is not that complex (and we don't mean that in a sarcastic way). If you look closely you will find out that all we are doing is making sure that bootloaders are installed in the proper places and don't overwrite the MBR. We hope this tutorial will help you to explore many more new operating systems, including countless Linux distributions and other wonderful UNIX flavours.

The complete guide to Perl – part 1

Swayam Prakasha introduces the basics to one of the most important open source scripting languages: Perl

Resources

Perl Operating systems such as UNIX and Linux have Perl already installed. Should it not be installed on your machine, go to the Perl homepage and follow the instructions there.
http://www.perl.org/get.html

Perl is considered to be very powerful and adaptable scripting language. It is widely used in automating system administration tasks. It is one of the most popular languages if we need to do some CGI programming in the web sphere. Another area where Perl is heavily used is in generating reports. **Perl comes with a wide set of features and some of them are listed below:**

a. It is an interpreted language.
b. It is an object-oriented language.
c. It's extremely useful for text processing.
d. As mentioned previously, it is very popular for web pages.

We typically call a Perl program a script and it can be created using any text editor. It is usual to save these files with a '.pl' extension.

The fact that a Perl program can run with little or no modification on many different platforms is another reason for its popularity.

01 Our first Perl program

All Perl programs start with '#' followed by the path where we have the Perl binary. This is pretty much required so that the script can be executed directly (**Fig 1**).

02 Executing a Perl script

Once we have a Perl program, use the chmod command to make the script executable. A Perl script can be executed in two ways (**Fig 2**). Note that the .pl extension is not mandatory.

03 Echo vs print statements

Unlike the echo command in Bash scripts, Perl's print command doesn't automatically send a carriage return and line feed. If you forget the \n sequence, the next print command will start on the same line. Also note that all Perl statements must end with a semicolon.

```
[root@centos1 perl]# cat perl_test1.pl
#!/usr/bin/perl
print "Welcome to Linux User and Developer\n";


[root@centos1 perl]#
```

Fig 1 Our first Perl program Print a simple statement using Perl

```
[root@centos1 perl]# ls -l perl_test1.pl
-rw-r--r-- 1 root root 64 Dec 21 09:03 perl_test1.pl
[root@centos1 perl]#
[root@centos1 perl]# chmod +x perl_test1.pl
[root@centos1 perl]#
[root@centos1 perl]# ls -l perl_test1.pl
-rwxr-xr-x 1 root root 64 Dec 21 09:03 perl_test1.pl
[root@centos1 perl]#
[root@centos1 perl]# perl perl_test1.pl
Welcome to Linux User and Developer
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test1.pl
Welcome to Linux User and Developer
[root@centos1 perl]#
```

Fig 2 Executing a Perl script How to execute a Perl script

04 Perl variables

By using a variable in a Perl script, we can reference a numeric or a character value. The beauty of Perl is that it figures out by context whether the value should be treated as a number or a character string and will even perform character-to-numeric value conversions when necessary. When we are assigning or accessing the value of a variable, we need to prefix it with a dollar sign. **Following are some of the examples of variable assignments.**

```
$VAR = 100; (assigning a numeric value)
$ARRAY = "welcome to my world"; (assigning a character string)
```

05 Perl variables – continued

In a Perl script, working with variables is fairly straightforward given that it is not necessary to define and allocate them (**Fig 3**). This saves a lot of time as there is no need to release the memory occupied by them.

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in iT for several years, concentrating on areas such as operating systems, networking, network security, eCommerce, and LDAP and web servers

```
[root@centos1 perl]# cat perl_test2.pl
#!/usr/bin/perl

$number=10;
print $number;

$message="\nperl script - variable assignment\n";
print $message;
[root@centos1 perl]#
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test2.pl
10
perl script - variable assignment
[root@centos1 perl]#
```

Fig 3 Perl variables Learn more about Perl variables

```
[root@centos1 perl]# cat perl_test3.pl
#!/usr/bin/perl

@Weekday =  ("Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun");

foreach $i (@Weekday)
{
        print"$i\n";
}
[root@centos1 perl]# ./perl_test3.pl
Mon
Tues
Wed
Thu
Fri
Sat
Sun
[root@centos1 perl]#
```

Fig 4 Types of Perl variables A look at Perl array variables

06 Types of Perl variables

There are 3 types of Perl variables:

a. Scalar variables
b. Array variables
c. Hashes or associate arrays

Scalar variables are simple variables that contain a single element. The single element could be a number, a string or a reference to an object.

Arrays are ordered list of scalars. Note that an array can hold an unlimited number of elements. An array's name begins with the '@' character (**Fig 4**).

Hashes or associate arrays have a group or pair of elements – a key and associated data.

07 Quotation marks in a Perl script

There are several distinct ways to use quotation marks in a Perl script. Let us take a look at the differences among single quotation marks, double quotation marks, backticks and the backslash character.

Single quotation marks will always give you exactly what's inside the quotation marks. We use double quotation marks when we need to assign a string that contains special characters that need to be interpreted.

The backslash is used to escape a single character (such as $ or *) that might otherwise be treated as a special character.

Finally, we use backticks to indicate that the string is a Linux command that should be executed.

```
[root@centos1 perl]# cat perl_test4.pl
#!/usr/bin/perl

$Array = "LU&D";

print 'Welcome to $Array\n';

print "\nWelcome to $Array\n";

$my_dir =  `pwd`;
print "You are currently in $my_dir\n";
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test4.pl
Welcome to $Array\n
Welcome to LU&D
You are currently in /root/perl

[root@centos1 perl]#
```

Quotation marks in Perl – a script and its execution

08 Predefined Perl variables

There are a lot of special variables in Perl that you need to know. These predefined variables are of scalar, array or hash type. One of the most important is a scalar variable $_.

```
[root@centos1 perl]# cat perl_test5.pl
#!/usr/bin/perl

$_= "Welcome to my tutorial\n";

print;
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test5.pl
Welcome to my tutorial
[root@centos1 perl]#
```

How predefined variables work in Perl

09 Arithmetic Perl operators

Arithmetic Perl operators are used in various arithmetic operations. Perl supports an assortment of arithmetic operators and they are used for arithmetic operations on one or more operands. These operators are used for addition, subtraction, negation, exponentiation, modulus, multiplication, division, auto increment and auto decrement.

```
[swayam@centos1 ~]$ cat perl_test6.pl
#!/usr/bin/perl

$VAR1 = 100;
$VAR2 =  55;

$VAR_Add = $VAR1 + $VAR2;
print "Sum is $VAR_Add\n";

$VAR_Sub = $VAR1 - $VAR2;
print "Diff is $VAR_Sub\n";

$VAR_Mod = $VAR1 % $VAR2;
print "Remainder is $VAR_Mod\n";
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test6.pl
Sum is 155
Diff is 45
Remainder is 45
[swayam@centos1 ~]$
```

Various arithmetic operators in Perl

10 Numeric relational operators

The numeric relational Perl operators compare two numbers and determine the validity of a relationship. Some such operators are '<', '>', '!=', '<=' etc. One important numeric relational operator is '< = >' and it is called 'numeric comparison'. This binary operator returns -1, 0, or 1, if the left operand is less than, equal to or greater than the right one, respectively.

```
[swayam@centos1 ~]$ cat perl_test7.pl
#!/usr/bin/perl

$VAR1 = 100;
$VAR2 =  55;

if ($VAR1 > $VAR2)
{
print "We are sure\n";
}

$VAR = $VAR1 <=> $VAR2;
print "$VAR\n";
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test7.pl
We are sure
1
[swayam@centos1 ~]$
```

Operators for numeric comparison

11 Logical operators

The numeric logical Perl operators are generally derived from Boolean algebra and they are mainly used to control program flow, finding them as part of an if, a while or some other control statement. Some of these operators are ! (negation), && (and), || (or), and ? (conditional operator) (**Fig 5**).

12 Bitwise operators

In Perl, numeric bitwise operators are similar to the logical operators, but they work on the binary representation of data. They are used to change individual bits in an operand. It is important to note here that both operands associated with bitwise operators are integers. The shift left operator (<<) shifts the bits to the left while the shift right operator (>>) shifts the bits to the right. We also have operators (such as & and |) that set a bit to 1 or 0 based on a specific condition.

```
[swayam@centos1 ~]$ cat perl_test9.pl
#!/usr/bin/perl

$VAR1 = 4 << 3;
print "$VAR1\n";

$VAR2 = 27 >> 3;
print "$VAR2\n";

[swayam@centos1 ~]$ ./perl_test9.pl
32
3
[swayam@centos1 ~]$
```

```
[swayam@centos1 ~]$ cat perl_test8.pl
#!/usr/bin/perl


$VAR = (23 == 23) ? "EQUAL\n" : "NOT EQUAL\n";
print $VAR;
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test8.pl
EQUAL
[swayam@centos1 ~]$
```

Fig 5 Logical operators A look at logical operatorsvariables

Learn more on bitwise operators

13 String relational operators

In Perl, the string relational operators are used to compare two strings and determine the validity of a relationship. Some of these operators are 'lt' (less than), 'le' (less than or equal to), 'gt' (greater than), 'ge' (greater than or equal to) and 'cmp' (comparison).

```
[swayam@centos1 ~]$ cat perl_test10.pl
#!/usr/bin/perl

($Val1, $Val2) = ("ABC", "ABX");

if ($Val1 lt $Val2)
{
print "$Val1 is less thann $Val2\n";
}

$Val = $Val1 cmp $Val2;
print "$Val\n";
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test10.pl
ABC is less thann ABX
-1
[swayam@centos1 ~]$
```

Understanding string relational operators

14 Reference operator in Perl

We call reference the scalar value that contains a memory address. In order to reference a specific variable, we use this operator (denoted by \). **The following sample script illustrates how to use this operator.**

```
$var="Welcome to LUD!";
$ref_v=\$var;
print $$ref_v, " (expected Welcome
to LUD!)\n";
```

15 Pattern binding operator

The pattern binding operator is a binary operator and is used to bind a string expression to a pattern match. The string which is intended to bind is put on the left, meanwhile the operator itself is put on the right.

```
[swayam@centos1 ~]$ cat perl_test11.pl
#!/usr/bin/perl

$VAL = "UNIX and LINUX";
 if($VAL =~ m/LINUX/) {
   print "YES\n";
 }

[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test11.pl
YES
[swayam@centos1 ~]$
```

A close look at the pattern binding operator

"In Perl, numeric bitwise operators are similar to the logical operators, but they work on the binary representation of data. They are used to change individual bits in an operand"

16 Perl lists

In Perl, a list is a group of scalars used to initialise an array or a hash. The elements of a list can be numbers, strings or any other types of scalar data. We use a numerical index to access the elements of a list. **The following is a simple example of a list.**

```
$Var1 = “Hello guys”;
(10, 100, ‘two numbers’, $Var1,
10.10)
```

17 Perl ‘qw’ operator

One can use the qw (known as ‘quote word’) operator to define the Perl lists (**Fig 6**). This operator will help you avoid writing too many quotation marks. But in cases where we have embedded whitespaces (spaces, tabs or newlines) in some list elements, we will not be able to use this operator.

18 Printing the list elements

As mentioned earlier, one can access the list elements by using an index (it is also known as a subscript).

```
[swayam@centos1 ~]$ cat perl_test13.pl
#!/usr/bin/perl

$Var1 = "Hello guys\n";
@LIST = (10, 100, 'two numbers', $Var1, 10.10);

print "$LIST[3]";
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test13.pl
Hello guys
[swayam@centos1 ~]$
```

Go and print elements of a list

```
[swayam@centos1 ~]$ cat perl_test12.pl
#!/usr/bin/perl

print qw(developer leade architect manager)[3];
print "\n";
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test12.pl
manager
[swayam@centos1 ~]$
[swayam@centos1 ~]$
```

Fig 6 Perl ‘qw’ operator How to use the qw operator in Perl

```
[swayam@centos1 ~]$ cat perl_test14.pl
#!/usr/bin/perl

print (11..19, "-", abc..abf),"\n";
[swayam@centos1 ~]$
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test14.pl
11121314151617181 9-abcabdabeabf[swayam@centos1 ~]$
[swayam@centos1 ~]$
```

Fig 7 Range operator in a list How the range operator works

19 Range operator in a list

Another popular operator used in the Perl lists is known as the range operator and it is denoted by “..”. The range operator basically expands all the intermediate values between the left and right values. Let us see how this operator works in an example (**Fig 7**).

```
[swayam@centos1 ~]$ cat perl_test15.pl
#!/usr/bin/perl

@Week_days =  ("Mon", "Tue", "Wed", "Thurs");
$extra_day = "Fri";
$LEN = push(@Week_days, $extra_day);

print "Length after appending is $LEN\n";
[swayam@centos1 ~]$
[swayam@centos1 ~]$ ./perl_test15.pl
Length after appending is 5
[swayam@centos1 ~]$
[swayam@centos1 ~]$
```

Fig 8 What we can do with Perl arrays? Append elements to arrays in Perl

> “Plenty of operators are available in Perl and they make the life of system administrators much simpler”

20 More on Perl arrays

With Perl arrays, the variable name starts with the @ character. An element of the array variable starts with the $ character followed by the variable name and the array element index included in square brackets. **Take a look at the following example:**

```
@software_ladder = (“programmer”,
“lead engineer”, “architect”,
“senior architect”);
```

We can understand the entire array through @software_ladder and to get the first element, we use $software_ladder[0].

21 What we can do with Perl arrays?

One can use the push function to append elements to Perl arrays (**Fig 8**). If we need to sort the elements of Perl arrays, we can go for the sort function. One can change the order of the array elements by using the reverse function, which reverses the order of the elements stored in an array.

In this article, we took a look at some of the basic concepts in the Perl language. Please watch out for the next issue of **Linux User & Developer** where we plan to cover intermediate concepts such as Perl statements and built-in functions in Perl – including hash functions, string functions and plenty of other functions which are useful for system administrators in their day-to-day operations.

The complete guide to Perl – part 2

The second part of the tutorial covers Perl statements: built-in functions in Perl including hash functions, string functions and other functions which are useful for system administrators

Resources

Perl Operating systems such as UNIX and Linux have Perl already installed. Should it not be installed on your machine, go to the Perl homepage and follow the instructions there.
http://www.perl.org/get.html

As with any other programming language, the statements are one of the most important topics in Perl. Generally, statements are used to process or evaluate the expressions. It is natural to expect these statements to return values and these values can be used to process or evaluate other statements. In its simplest form, one can consider a Perl program as a sequence of statements and expressions, from the beginning to the end of the program. A Perl statement ends with a semicolon (;) and this character is used to inform the interpreter that the statement is complete. It is common to put several statements in the same line within a script, with each statement ending with a semicolon.

01 Perl statement

Here are some examples of Perl statements:

```
7+8;
$a = 2+5;
$c = $a; $b = 7-3;
```

The first statement is a simple addition and the result of the statement will vanish (as it is not assigned to any variable). In the second one, we have captured the result of the statement in a variable. The third example features two statements on the same line.

02 Perl blocks

We can group several Perl statements in a block. It is common to enclose a block with a pair of curly brackets. It is important to understand here that every statement in a block is evaluated by Perl as an individual statement. After the last statement of a block, we can omit the semicolon, but it is good practice to end the last statement with a semicolon.

Let us take a look at a sample script to understand this concept.

```
#!/usr/bin/perl
@NUM = (1,2,3,4);
foreach $i(@NUM) {
$i++;
print “$i”;
}
print ”\n”;
```

The screenshot above (**Fig 1**) explains the execution and output of this script.

```
[root@centos1 perl_scripts]# cat perl_test1.pl
#!/usr/bin/perl

@NUM = (1,2,3,4);

foreach $i(@NUM) {

$i++;
print "$i";
}

print "\n";
[root@centos1 perl_scripts]# ./perl_test1.pl
2345
[root@centos1 perl_scripts]#
```

Fig 1 Perl blocks Script to understand blocks

03 Perl compound statements

When we group one or more blocks and expressions in Perl, we get a compound statement. To define a compound statement, we may have more complex examples such as blocks nested in other blocks and so on.

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in iT for several years, concentrating on areas such as operating systems, networking, network security, eCommerce, and LDAP and web servers

04 Perl control statements

There is an important category of Perl statements and they are the control statements. **One can divide the control statements into two categories...**

a. Conditional statements
b. Loop statements

Conditional statements use an expression to see if a condition is met or not. Loop statements check a condition and execute the statements included in curly brackets until that condition is evaluated true/false.

05 Conditional statements – if-else

The expressions in a conditional statement are evaluated in a Boolean context. One can have various clauses in a conditional statement and the most commonly used are 'if' and 'else' (**Fig 2**). One can also use 'switch' and 'unless' Perl statements. The idea behind if-else is that if the conditional statement is not met then do something else.

06 Conditional statements – elsif

In Perl, elsif statements (**Fig 3**) test another conditional statement before executing code. In this way, one can test multiple conditions before the script continues. The following script can be used to understand this concept.

```
[root@centos1 perl_scripts]# cat perl_test2.pl
#!/usr/bin/perl

$Magazine = "LU&D";

if ($Magazine eq "LU&D") {
       print "Good, go and visit LU&D\n";
}
else
{
       print "It is not LU&D\n";
}
[root@centos1 perl_scripts]#
[root@centos1 perl_scripts]# ./perl_test2.pl
Good, go and visit LU&D
[root@centos1 perl_scripts]#
```

Fig 2 Conditional statements – if-else To understand the if-else clause

```
[root@centos1 perl_scripts]# cat perl_test3.pl
#!/usr/bin/perl

$var =10;

if ($var == 5) {
       print "Variable value is 6\n";
}
elsif ($var == 10) {
       print "Variable value is 10\n";
}

[root@centos1 perl_scripts]# ./perl_test3.pl
Variable value is 10
[root@centos1 perl_scripts]#
```

Fig 3 Conditional statements – elsif To understand the elsif construct

```
#!/usr/bin/perl
$var =10;

if ($var == 5) {
       print "Variable value is
6\n";
}
elsif ($var == 10) {
       print "Variable value is
10\n";
}
```

07 Perl switch statement

The Perl 'switch' statement is an alternative way to write conditional statements. The syntax of this switch statement is given in the following example...

```
switch(variable) {
   case 1:
        statement_1";
        break,
   case 2:
        statement_2";
        break,
   default:
        default_statement
}
```

The 'break' statement is used to leave the Perl switch statement and continue with the code that follows the switch statement.

08 Perl loop statements

Sometimes we need to execute a code block more than once. A loop statement allows one to execute a set of Perl statements enclosed in braces repeatedly until an expression is evaluated as true/false. **Loop statements follow a three-step execution process:**

a. Evaluate an expression in a Boolean context.
b. If the expression is evaluated as true/false, execute the next block of statements.
c. Repeat step a.

```
[root@centos1 perl_scripts]# cat perl_test6.pl
#!/usr/bin/perl

for ($var = 10; $var >= -10; $var--) {

       print "$var";
}
       print "\n";

[root@centos1 perl_scripts]# ./perl_test6.pl
10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
[root@centos1 perl_scripts]#
```

Fig 4 Perl loop statement – for Script to understand the 'for' loop

09 Perl loop statements – while and do-while

The 'while' statement executes the code block repeatedly as long as a conditional expression is true. Here the expression is tested before the first iteration of the loop. One can use a do-while loop statement when there is a need to execute a piece of code at least once and then iterate consecutively while the condition is true.

```
[root@centos1 perl_scripts]# cat perl_test5.pl
#!/usr/bin/perl

$name = "Linux";

while ($name eq "UNIX") {
       print "It is UNIX\n";
}
       print "It is Linux\n";

[root@centos1 perl_scripts]# cat perl_test4.pl
#!/usr/bin/perl

$var =1;

do{

       ++$var;
       print "$var";
} while $var < 5;

print "\n";

[root@centos1 perl_scripts]#
```

■ Understanding Perl loop statements

10 Perl loop statement – for

A 'for' loop executes a given set of statements for a certain number of times (**Fig 4**). In a 'for' statement, the conditional expression consists of three parts: initialisation, condition and increment. These are delimitated by a semicolon symbol. The condition will be an expression that will be evaluated and if it is true, the block will be executed. Otherwise, the 'for' loop will be terminated.

11 Perl loop statement – foreach

This construct operates over a list of values or an array. It is a very useful control looping statement and is used whenever we need to perform some actions on the individual elements of an array or a list (**Fig 5**, overleaf).

12 Perl loop statements – until and do-until

The 'until' construct is similar to the while construct. The only difference is that it executes the block only if the expression is false, and the expression will be tested before the first iteration. There is another construct, do-until, and it operates similar to do-while with a single difference: it iterates only if the conditional expression is evaluated as false.

13 Built-in functions in Perl

Perl provides several built-in functions and you do not need to declare them when you are using them in your script. As expected, when Perl meets the name of a function in a Perl script, it stops executing the sequential lines of code and continues with the execution of the function code. After finishing the execution of the function, Perl returns at the point from where the function was called and continues with the current code that follows the called function.

14 String functions

Various string functions available in Perl will enable you to perform different operations on strings and scalars. Some of these functions are useful for obtaining a string's length, getting the position of a substring in a string, removing and returning the last character from a string etc.

```
[root@centos1 perl_scripts]# cat perl_test7.pl
#!/usr/bin/perl

$my_sum = 0;

@numbers = (2,4,6,8);
foreach (@numbers) {
        $my_sum += $_;
}
        print "Sum is $my_sum\n";

[root@centos1 perl_scripts]#
[root@centos1 perl_scripts]# ./perl_test7.pl
Sum is 20
[root@centos1 perl_scripts]#
```

Fig 5 Perl loop statement – foreach Using foreach to sum the elements of a list

15 A popular string function – chomp

This string function is useful in removing trailing new lines. Let us take a look at a sample script to get a better understanding of this function (**Fig 6**)...

```
#!/usr/bin/perl
print "Enter a user name: ";
$name = <STDIN>;
chomp $name;
print „You entered : $name\n";
```

```
[root@centos1 perl_scripts]# cat perl_test8.pl
#!/usr/bin/perl

print "Enter a user name: ";
$name = <STDIN>;
chomp $name;
print "You entered : $name\n";

[root@centos1 perl_scripts]# ./perl_test8.pl
Enter a user name: Mike Pieterson
You entered : Mike Pieterson
[root@centos1 perl_scripts]#
```

Fig 6 A popular string function – chomp Script and execution for the chomp function

> "Several built-in list functions are available in Perl"

This script will read from the keyboard and the value read will be stored in a variable name. This variable will have a new line character at the end and the string function chomp will take care of it.

16 List functions

Several built-in list functions are available in Perl and by using them, one will be able to manipulate lists. **The popular ones are grep and join.**

a. grep – This function is used as a filter. Based on the evaluation of an expression, this function is used to extract elements of a list.

Let us take a look at the following script:

```
#!/usr/bin/perl
@Employees = ('Mike', 'Peter',
'Shyam', 'Mat');
@Result = grep(/^M/, @Employees);
```

Here, we are looking for values that start with a capital M. And as expected, the value of @Result will be Mike and Mat.

b. join – This function is useful for merging any number of strings. It is used to concatenate the elements of an array or a list into a string, using a specified separator (**Fig 7**).

17 Array functions in Perl

Perl provides many built-in functions to manipulate arrays. It is important to note here that every time we call a Perl array function, we need to differentiate between a simple array and a multidimensional array. Some of the important and widely used array functions are 'shift', 'push', 'pop' and 'unshift'.

The shift function removes and returns the first element of an array, thereby reducing the number

```
[root@centos1 perl_scripts]# cat perl_test9.pl
#!/usr/bin/perl

$my_string = join( "--", "Linux", "User", "Developer");
print "Here is the joined string: $my_string\n";
[root@centos1 perl_scripts]#
[root@centos1 perl_scripts]# ./perl_test9.pl
Here is the joined string: Linux--User--Developer
[root@centos1 perl_scripts]#
```

Fig 7 List functions Understanding the join function with an example

```
[root@centos1 perl_scripts]# cat perl_test10.pl
#!/usr/bin/perl

@days = qw(Sun Mon Tue Wed Thu Fri);
$weekly_off = shift(@days);

print "Weekly off is on $weekly_off, Working days are @days\n";
[root@centos1 perl_scripts]#
[root@centos1 perl_scripts]#
[root@centos1 perl_scripts]# ./perl_test10.pl
Weekly off is on Sun, Working days are Mon Tue Wed Thu Fri
[root@centos1 perl_scripts]#
```

Fig 8 Array functions in Perl Take a look at how the shift function works

of array elements by one. **The following sample script illustrates the use of this function.**

```
#!/usr/bin/perl
@days = qw(Sun Mon Tue Wed Thu Fri);
$weekly_off = shift(@days);
print “Weekly off is on $weekly_off,
Working days are @days\n”;
```

(**Fig 8**)

18 Perl hash functions

Perl supplies numerous functions that can be used to manipulate hashes. Some of the important ones are ‘delete’, ‘keys’ and ‘values’.

Let us take a look at how the keys function works. This function is used to get the indices of an array or keys of a hash. **Let us understand this with a sample script:**

```
[root@centos1 perl_scripts]# cat perl_testt12.pl
#!/usr/bin/perl

%hash = (Manager => 'John', Leader => 'Scott',
          Engineer => 'Andrew', Trainee => 'Peter');

print "The designations are: ";
foreach $key (keys % hash) {
  print "$key ";
}
print "\n";


[root@centos1 perl_scripts]#
[root@centos1 perl_scripts]# ./perl_testt12.pl
The designations are: Leader Engineer Manager Trainee
[root@centos1 perl_scripts]#
```

Fig 9 Perl hash functions Extracting keys from a hash

> **“Perl supplies numerous functions that can be used to manipulate hashes. Some of the important ones are ‘delete’, ‘keys’ and ‘values’”**

```
#!/usr/bin/perl
%hash = (Manager => ‘John’, Leader
=> ‘Scott’,
          Engineer => ‘Andrew’,
Trainee => ‘Peter’)

print “The designations are: “;
foreach $key (keys % hash) {
print “$key “;
}
print “\n”;
```

19 Other functions available in Perl

There are a number of miscellaneous functions available in Perl and they can be used to perform various operations on strings, arrays and hashes. The important miscellaneous functions are ‘defined’, ‘scalar’ and ‘undef’. The ‘defined’ function allows us to check whether a function or a variable is defined or not. We can use a ternary conditional operator to see if a variable is defined or not.

In this article, we took a look at various Perl statements and got an idea on how to use them effectively. We also learnt about an important aspect of Perl – built in functions – and studied them with sample examples. We will explain more features of Perl in the next issue of **Linux User & Developer**.

The complete guide to Perl – part 3

In the final part of his Perl tutorial series, Swayam Prakasha covers file management, file handles, Perl scripting for databases and more besides...

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in iT for several years, concentrating on areas such as operating systems, networking, network security, eCommerce, and LDAP and web servers

In the last two issues of Linux User & Developer, we studied various basics and intermediate features of Perl. We had a look at various features and functionalities of Perl and understood why Perl is considered as the de facto standard scripting language for automating system administration tasks. Readers are advised to go through those two tutorials to obtain a good knowledge on several aspects of Perl before continuing here.

In the first part of our tutorial, we examined various components of a Perl script. We had a good look at Perl variables and found out that these variables are very useful in the day-to-day scripting activities of system administrators. Then we described various operators and explored their usefulness with sample scripts. We also had a look at lists and arrays and our focus was to understand the various operations that can be performed on lists and arrays. In the second tutorial, we studied various Perl statements, built-in functions in Perl – including hash functions and string functions – and saw how they can be used effectively by system administrators.

As mentioned in the first tutorial, readers are expected to have some knowledge of UNIX / Linux operating systems, not to mention some exposure to scripting languages.

Operating systems such as UNIX and Linux already have Perl installed. If it is not installed on your machine, however, please make sure that you install it from a reliable source.

File handling is an important aspect when it comes to Perl scripting. The basics of file handling in Perl are pretty straightforward – we associate a file handle with a file and later we use various operators and functions (available in Perl) to read and update the data stored within the data stream associated with the file handle. In Perl, a file handle is an internal structure which basically associates a file with a name. It is important to note here that all file handles are capable of read-and-write access. When we associate a file handle, we are also able to specify the mode in which a file handle can be opened.

> **"In Perl, a file handle is an internal structure which basically associates a file with a name... When we associate a file handle, we are also able to specify the mode in which a file handle can be opened"**

Resources

Perl Operating systems such as UNIX and Linux have Perl already installed. Should it not be installed on your machine, go to the Perl homepage and follow the instructions there.
http://www.perl.org/get.html

01 Opening a file

The function 'open' can be used to open any new or existing file in Perl. **If we need to open a file (say sample.txt) in read-only mode, then the syntax will look like**

```
open(FILE_HANDLE, "<sample.txt")
```

The following example shows how to use this function to open a file

```
#!/usr/bin/perl
open(FILE_HANDLE, "<sample.txt");
while (<FILE_HANDLE>)
{
        print "$_";
}
```

And we have captured the results of this in the screenshot on the next page (**Fig 1**).

02 Understanding sysopen function

The function 'sysopen' is very similar to the 'open' function. The only difference is that sysopen uses the system open() function. **For example, if we need to open a file in read-write mode, we can go for the following format**

```
sysopen(FILE_HANDLE, "sample.txt"
O_RDWR);
```

03 Closing a file

Closing a file is very straightforward – by closing a file, we disassociate the file handle from the corresponding file. One can use the 'close' function for this. **The syntax is as follows:**

```
close
```

Or

```
close FILE_HANDLE
```

If no file handle is specified, then 'close' closes the currently selected file handle.

```
[root@centos1 perl]# cat sample.txt
This is a sample file to test open functionality.
[root@centos1 perl]#
[root@centos1 perl]# cat perl_test6.pl
#!/usr/bin/perl

open(FILE_HANDLE, "<sample.txt");

while (<FILE_HANDLE>)
{
        print "$_";
}
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test6.pl
This is a sample file to test open functionality.
[root@centos1 perl]#
[root@centos1 perl]#
```

Fig 1 Opening a file Open an existing file in Perl

04 Read and write functionalities

Once we have an open file handle, we will be able to read and write. Reading and writing data into a file can be done in many ways. **Take a look at the following sample script**

```
#!/usr/bin/perl
print "Enter a name here\n";
$my_name = <STDIN>;
print "You entered $my_name\n";
```

The main method of reading the information is to use the <FILE_ HANDLE> operator. The following screenshot displays the output of the above script.

```
[root@centos1 perl]# cat perl_test8.pl
#!/usr/bin/perl

print "Enter a name here\n";
$my_name = <STDIN>;

print "You entered $my_name\n";
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test8.pl
Enter a name here
MIKE
You entered MIKE

[root@centos1 perl]#
```

■ Reading from a file handle

Sometimes, we will be interested in importing all lines from a file into an array. **The following sample script illustrates how it can be done**

```
#!/usr/bin/perl
open(MY_HANDLE, "<sample.txt");
@LINES = <MY_HANDLE>;
print "@LINES";
close (MY_HANDLE);
```

The screenshot on the right (**Fig 2**) displays the execution of the above script.

Other functions such as read, getc and print can also be used for reading / writing into a file in Perl.

Fig 2 Read and write functionalities Reading from a file

```
[root@centos1 perl]# cat sample.txt
This is a sample file to test open functionality.
And we are testing the reading functionality also.
[root@centos1 perl]#
[root@centos1 perl]#
[root@centos1 perl]# cat perl_test7.pl
#!/usr/bin/perl

open(MY_HANDLE, "<sample.txt");
@LINES = <MY_HANDLE>;
print "@LINES";
close (MY_HANDLE);
[root@centos1 perl]#
[root@centos1 perl]#
[root@centos1 perl]# ./perl_test7.pl
This is a sample file to test open functionality.
 And we are testing the reading functionality also.
[root@centos1 perl]#
```

```
[root@centos1 perl]# cat sample1.txt
This is a sample file to test open functionality.
And we are testing the reading functionality also.
[root@centos1 perl]#
[root@centos1 perl]# cat perl_test9.pl
#!/usr/bin/perl

open(MY_HANDLE1, "<sample1.txt");

open(MY_HANDLE2, ">sample2.txt");

while (<MY_HANDLE1>)
{
        print MY_HANDLE2 $_;
}

close(MY_HANDLE1);
close(MY_HANDLE2);

[root@centos1 perl]# ./perl_test9.pl
[root@centos1 perl]# cat sample2.txt
This is a sample file to test open functionality.
And we are testing the reading functionality also.
[root@centos1 perl]#
```

Fig 3 Copying files Copying files in Perl

05 Copying files

One way of copying files is to open an existing file, read it line by line and then generate the second file. **The following script illustrates this**

```
#!/usr/bin/perl
open(MY_HANDLE1, "<sample1.txt");
open(MY_HANDLE2, ">sample2.txt");
while (<MY_HANDLE1>)
{
        print MY_HANDLE2 $_;
}
close(MY_HANDLE1);
close(MY_HANDLE2);
```

Execution of the above script and contents of the new file are captured in the screenshot at the top of the page (**Fig 3**).

06 Renaming and deleting a file

We can use the 'rename' function to rename an existing file. **The following script shows how to use this function**

```
#!/usr/bin/perl
rename ("/root/perl/sample1.txt","/
root/perl/test.txt");
```

The function 'unlink' can be used to delete a file. We have a sample script to illustrate this in the screenshot overleaf (**Fig 4**).

07 Locating the position within a file

If we are interested in knowing the current position of a file, then we can use the 'tell' function. The 'lseek' function is used to point to a particular position inside the file.

When we use the tell function (like tell FILE_HANDLE), it returns the position of the file pointer (in bytes). **Use of lseek requires three parameters – an example is given below**

```
lseek(FILE_HANDLE, 128,0);
```

Here the second parameter sets the file pointer to the 128th byte in the file (relative to the start of the file as we have specified the third parameter as 0).

08 Error handling in Perl

It is always very important to handle errors in our Perl scripts. There are different ways by which one can identify and trap these errors. Once trapped, errors need to be handled properly. **Let us take a look at the following script**

```
if (open (MY_HANDLE, "<sample.txt")
)
{
   //do something
}
else
{
   die "Error - not able to open the
file $!";
}
```

The variable $! returns the actual error message.

We also have a 'warn' function in Perl that raises a warning (by printing a message to standard output). Note that the 'die' function works like warn – the only difference is that die also calls exit. Thus this function has the effect of terminating the execution immediately.

09 Sockets in Perl

As we know, a socket is a mechanism of creating a virtual duplex connection between processes. Various socket calls (such as socket, bind, listen, accept, connect etc) are applicable in Perl and one can use them for both connection-oriented and connectionless network communication. **A call to the socket function will look like**

```
use Socket;
socket( SOCKET, DOMAIN, TYPE,
PROTOCOL );
```

In the case of a connection-oriented client-server, the following steps need to be taken care of...

At server end

a. Create a socket
b. Bind the socket to a port address
c. Listen to the socket at the port address
d. Accept the client connection

At client end

a. Create a socket
b. Connect to the server

10 String matching

Though we have used string matching earlier, it is good to take another look at it. Perl provides powerful string manipulation facilities. System administrators heavily use 'regular expressions' for string matching.

Let us consider an example

```
#!/usr/bin/perl
$statement = "I am very good in the
morning today";
$statement =~ /good/;
print $?;
print "\n";
```

In the above example, the expression '$statement=~/the/;' is true as the string 'good' appears in the variable $statement. The screenshot on the next page (**Fig 5**) explains this.

11 More on regular expressions

One can see plenty of special characters in a regular expression. Because of this, regular expressions are becoming very powerful. Some of these special characters and their meaning are given in the table on the next page (**Table 1**).

12 Substitution in Perl

Based on the matches identified by regular expressions, we can make substitutions in Perl. This can be accomplished by using the 's' function. **The following expression replaces an occurrence of 'linux' by 'LINUX' in the string $Statement**

```
$Statement =~ s/linux/LINUX/
```

To make a global substitution, we can go for the 'g' option, as shown below

```
$Statement =~ s/linux/LINUX/g
```

13 Database management in Perl

Perl is capable of running SQL and MySQL queries. These queries include inserts, deletes, updates, selects etc. All these are done through a module known as DBI. DBI stands for 'database interface' in Perl. It defines a

```
[root@centos1 perl]# ls
perl_test1.pl  test.txt
[root@centos1 perl]#
[root@centos1 perl]# cat perl_test1.pl
#!/usr/bin/perl

unlink ("/root/perl/test.txt");


[root@centos1 perl]#
[root@centos1 perl]# ./perl_test1.pl
[root@centos1 perl]#
[root@centos1 perl]# ls
perl_test1.pl
[root@centos1 perl]#
```

Fig 4 Renaming and deleting a file Delete an existing file

> **"Perl provides several interesting file management functions and it is up to the system administrator to make the best use of them"**

Table 1: Special characters and their meanings

Special character	Meaning
.	Any single character except a new line
^	The beginning of the line or string
$	The end of the line or string
*	Zero or more of the last character
+	One or more of the last character
?	Zero or one of the last character

set of methods, variables and conventions that provide a consistent database interface independent of the actual database being used. First we need to instruct Perl to use specific modules and this is illustrated below.

```
#!/usr/bin/perl

#Here are the Perl modules that we
need to use
use DBI;
USE DBD::mysql;
```

These specific modules allow us to call upon functions specific to working with any database platform, including MySQL.

> **"The DBI module defines a set of methods, variables and conventions that provide a consistent database interface independent of the actual database being used"**

14 Connecting to the database

For connecting to a database, one can use 'connect'. The connect string takes three parameters: a data source name, username and password. Please understand the format of data source name before attempting a connection to a database. **Four pieces of information are needed to make up a data source name:**

a. Name of SQL platform
b. Database name
c. Host name
d. Port number

The 'connect' function will return 'true' on success; 'untrue' otherwise. Also, DBI will place an error message in the package variable $DBI::errstr. One can also use the die() function to terminate the script upon a connection failure. A simple script for establishing a connection to a database is shown below.

```
useDBI;
my $dbh=DBI->connect('dbi:
DBMaker:dbtest','my_user', 'my_
userpassword',);
```

15 CGI programming with Perl

Perl is popularly used in CGI (Common Gateway Interface) programming. All CGI programs to be executed are kept in a specific directory and these programs have an extension of .cgi. **We take a look at our first CGI program now:**

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "Hello, This is our first CGI
program!\n";
```

Please remember that CGI is simply a program called by the web server, in response to some action by a web visitor. **Follow these steps to see the results**

a. Save or upload the file into your web directory.
b. Make the file executable by changing the permissions.
c. Now go to your web browser and type the URL for your CGI program.

There are various environmental variables and all CGI programs will have access to these. **The following CGI program displays all these environmental variables.**

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<font size=+1>Environment</
font>\n";
foreach (sort keys %ENV)
{
  print "<b>$_</b>: $ENV{$_}<br>\n";
}
```

Thus, in this final part of our Perl tutorial, we took a look at various file management functionalities. We also looked at how Perl can be used in CGI (Common Gateway Interface) programming. Now it is time for you to go back and use all the features of Perl in your day-to-day operations.

```
[root@centos1 perl]# cat perl_test1.pl
#!/usr/bin/perl

$statement = "I am very good in the morning today";

$statement =~ /good/;

print $?;
print "\n";

[root@centos1 perl]#
[root@centos1 perl]# ./perl_test1.pl
0
[root@centos1 perl]#
[root@centos1 perl]#
```

Fig 5 String matching Understanding string matching in Perl

Supercharge KDE

We bring you the most useful tips and tricks to make your KDE ride more enjoyable – whether you are a user, administrator or developer. Plus a brief guide to installing KDE on Mac OS X

Introduction to KDE 4

KDE provides everything that today's modern desktop system does and more – much, much more. KDE is already included with all popular Linux distributions and UNIX variants. This includes all flavours of Linux (openSUSE, Fedora, Mandriva, Ubuntu/Kubuntu), BSDs (FreeBSD, NetBSD, OpenBSD), IBM AIX and Sun Solaris. We almost forgot to mention Mac OS X. OS X is also a UNIX variant; we should probably put it where we mentioned the BSD OS. There is also one other platform with which KDE works: Windows. KDE does not aim to provide the full desktop environment on Windows and Mac OS X, however, as they already have a capable desktop. Instead it provides KDE applications and its development libraries on these platforms.

We are not going to stress how cool KDE is – since you're reading this feature, we'll assume you already have a pretty good idea. Instead we will focus on how to make this cool desktop experience even better. We have something for everyone, whether you're a user, administrator or a KDE developer. Yes – even developers. Enough chit-chat, though, it's time to jump right in...

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Note

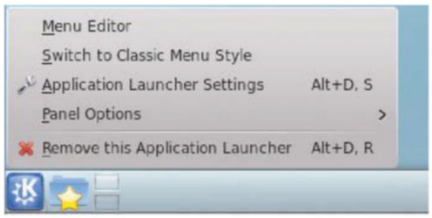
This feature is written using KDE 4.5.2 (installed as part of Kubuntu, which is on this month's disc for you to try). Most of the things discussed here may work with KDE 4.1 version onwards; we make no promises. Most of the changes are related to the way some menu items are placed or named. We highly recommend you to upgrade your KDE to the latest release (4.5 onwards) regardless of whether you want to follow this feature or not.

KDE user tips & tricks

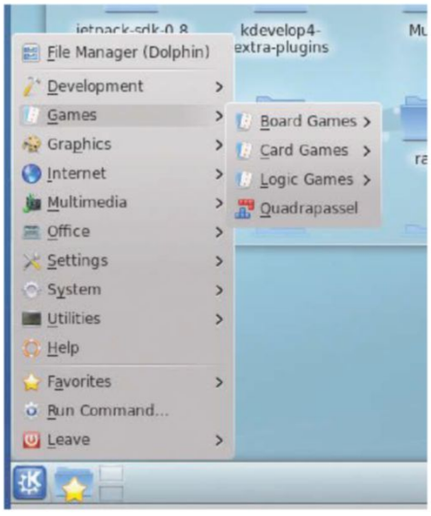
1. Launching applications the old way

KDE 4 introduces a new Kickoff-style application launcher. The new application launcher is innovatively designed to provide access to most of the important parts of KDE (such as favourites, recently used items) right from the Application Start menu. But you may not like it if you are all about getting to your application first. We're still in love with the classic menu style, and the chances are that you are too.

To switch to the classic menu, right-click the Start menu and select 'Switch to Classic Menu Style' from the context menu. Welcome back the good old classic application launcher!



Context Menu



Classic-style application launcher

2. More widgets!

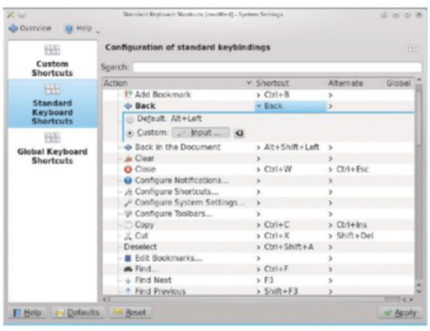
By default KDE ships with a set of Plasma widgets that include everything from Battery Monitor to Folder View. But you can get even more widgets right from your desktop. To get new widgets, right-click on the desktop, select 'Add Widgets', then 'Get New Widgets...'.

Alternatively, you can also search your distribution's package manager with text 'Plasma' to install new widgets.

It's not all about Plasma widgets, though, since KDE also supports Google Gadget Engine. Search the package manager for the term 'plasma-scriptengine-googlegadgets' to install the support for Google Gadgets. You can now use the same method to install Google Gadget widgets.

3. Modifying keyboard shortcuts

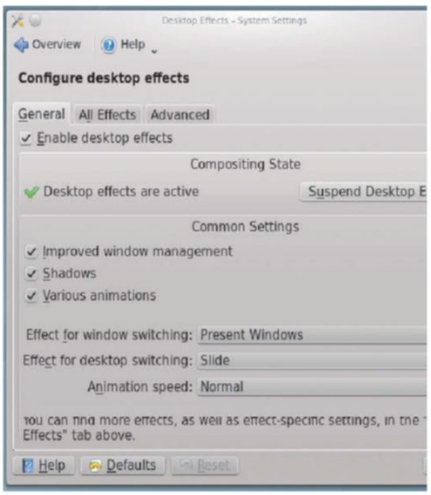
We all have a mouse or trackpad now, but nothing beats the speed and efficiency of using keyboard shortcuts to get something done. But keyboard shortcuts, like any other habit, are hard to change. Could you imagine trying to save a file using Alt+M when you are so used to Ctrl+S? Thankfully KDE makes it very easy to map the keyboard shortcuts to your own liking. Open System Settings (KDE) and select 'Shortcuts and Gestures'. The next screen allows you to create custom shortcuts, modify standard keyboard shortcuts and modify global keyboard shortcuts. Just click on any shortcut you want to modify, then select Custom, click on the Input button and enter type the key combination that you want to use as shortcut.



Modifying KDE keyboard shortcuts

4. 3D effects

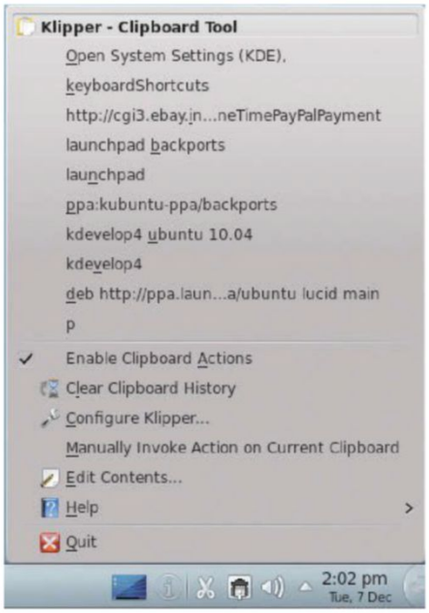
We hope you've already had the memo about 3D. It is not just for games any more, but also helps accelerate desktop productivity while offering an awesome look. No prizes for knowing KDE has built-in support for 3D. You can use 3D effects for various desktop events such as Window Switching, Desktop Switching, Window Minimize, Window Maximize, Window Move and so on. These effects not only look cool, but can also increases your productivity. The only catch is that your Linux system must have hardware-accelerated graphics enabled. For Nvidia and AMD/ATI graphics cards you'll need to install the proprietary drivers for best performance. To enable 3D effects on KDE desktop, open System Settings (KDE), select Desktop Effects (under 'Workspace Appearance and Behavior') and check Enable Desktop Effects. You can then modify additional settings to your liking.



Enabling desktop effects

5. Copy & pasting smartly

Copy and paste is an important functionality of any desktop system. But the traditional implementation in desktop systems is not very efficient. For example, if you have copied three items, you can only paste the last item you have copied. But there is much more you can do with KDE's copy-paste system called Klipper. You can paste from the clipboard history, modify clipboard contents before pasting, and use clipboard actions. Clipboard actions is probably one of the coolest features to have in the clipboard system. You can execute actions based on a MIME type and the text matches based on regular expressions. For example, you can configure Klipper to open a new email window when you copy an email address. Klipper should be available on all recent KDE systems – if not, search your distribution's package manager for Klipper and then run the command '$ klipper &' to start the Klipper system. Once Klipper is available, you can right-click the scissor icon available in the System Tray to configure it to your liking.



Klipper menu with clipboard history

6. Keeping the dirty habit

Lot of us have the habit of dumping folders and files on the desktop so we can quickly access them. With the introduction of KDE 4's Plasma desktop system, this ability was removed. A lot of people weren't too happy about it so the KDE team answered in the form of a widget called Folder View. Now, the Folder View is not a complete replacement, but it does show the files of a selected folder on the desktop. The main advantage is that you can use this view to show any folder you want. To put this widget on the desktop, right-click on the desktop, select Add Widgets, then select Folder View.

Folder View

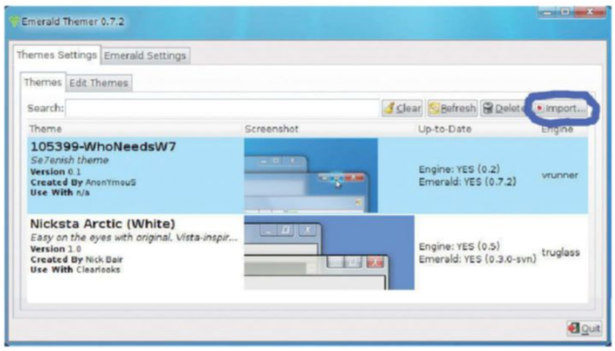


Fig 1 Using a different window manager
The Emerald theme manager running and displaying a Windows 7-like theme

7. Using a different window manager

KDE is a very modular system. This enables users to mix and match different components to make the complete desktop experience. Sometimes you can even replace KDE components with a third-party component. One such component is the KDE window manager, KWin. A window manager is a windowing system that runs on top of X Window. Without a window manager you won't be able to minimise, maximise, close, move or do anything at all with application windows. While KWin is very capable window manager, you may want to try window managers from other projects to get a different set of features integrated right into the KDE experience.

Every X Window manager supports the '--replace' command which allows the new window manager to take over the current window manager. If a window manager is buggy and crashes at times, you may not be able to work with your windows and you will need to start the window manager again.

One of the best window managers (outside of the KDE project) is Emerald (**Fig 1**). Emerald is part of the Compiz project which provides all sorts of different configurations and layouts of buttons, looks, title bars and frames. **To use Emerald, do the following:**

```
$ emerald --replace &
```

to revert back to KWin

```
$ kwin --replace &
```

8. Reset to the factory mode

If at some later point you believe that it is time to revert back to the KDE defaults, you can do so very easily by removing the KDE's personal configuration directory from your home directory.

`$ rm -r -v .kde4` OR...

`$rm -r -v .kde` (beware, you could be deleting the KDE 3 configuration directory; check your distribution's documentation first)

It is important to remember that deleting the complete configuration directory will cause all KDE applications to lose your personal data such as saved emails from KMail. If you want to reset the configuration for an individual KDE application, look into <KDE4 Home Config Directory>/share/apps and delete the required application. To delete a particular setting, you can delete individual settings files from <KDE4 Home Config Directory>/share/config.

9. Installing Compiz Fusion on KDE

KDE's built-in compositing effects are great, but sometimes you may want a more mature solution or simply more eye-candy to decorate your desktop. Compiz Fusion is a dedicated compositing window manager with full support for KDE and GNOME desktops. Installing Compiz Fusion in KDE is very easy. **There are two things to do...**

Disable the built-In effects: First we need to disable the KDE's built-in desktop effects so that they do not create problems with Compiz Fusion. Open KDE System Setting>Desktop Effects, then uncheck Enable Desktop Effects.

Installing Compiz Fusion: Search and install the following packages using the package manager:

1. compiz
2. compiz-kde4
3. compiz-manager
4. compiz-plugins-extra
5. compiz-plugins-main
6. libcompizconfig
7. compizconfig-settings-manager
8. python-ccm
9. python-compizconfig
10. simple-ccsm-kde
11. emerald

After installation, run the command 'simple-ccsm' and select a profile to get started with Compiz Fusion.

KDE administrator tips & tricks

1. Using KDE protocols to do more in KDE applications

Every file manager supports browsing files, but KDE applications (including Dolphin and Konqueror) go way further. Taking a page from the internet's book, KDE provides a protocol-based solution to browsing almost any supported protocol. For example, you can use ftp:// to point Dolphin to an FTP-based site, or you can use sftp:// to access files over SSH; you can even use ssh:// to open an SSH connection to the server in Konsole. These protocols also expand to non-network-related mechanisms such as zip://, which allows you browse inside ZIP archives, or accessing DVD discs (dvd://). To get a list of installed protocols, you can look inside /usr/share/kde4/services.

```
$ ls *.protocol
about.protocol  floppy.protocol
mmsu.protocol   smtp.protocol
aim.protocol    fonts.protocol
nepomuk.protocol        smtps.
protocol
akonadi.protocol        ftp.protocol
nepomuksearch.protocol  ssh.protocol
amarokitpc.protocol     ghelp.
protocol        network.
protocolsvn+file.protocol
amaroklastfm.protocol  groupwise.
protocol        nfs.protocol
svn+http.protocol
amarok.protocol groupwises.protocol
nntp.protocol   svn+https.protocol
......           .......
.......                  ......
```

■ Dolphin opening an SFTP connection to 192.168.1.2

2. Viewing log files

We all know where the logs are. But remembering where all the log files, including system and application log files, are and searching for appropriate information from those log files could be a daunting task. You can use the KDE app KSystemlog to view and analyse all the log files in one place. In addition to supporting system log files (such as System Log, Kernel Log, Daemon Log, Authentication Log, Xorg.log) the app supports various application-specific logs such as Apache Log, Postfix Log etc.

Logs are shown in colour-coded format for Information, Warnings and Errors. You can do a full text search on the log files or filter results using a particular field.

3. Accessing the terminal server

You can use KRDC to view or even control the desktop session on another machine that is running a compatible server. Most of the platform is supported via VNC (Virtual Network Computing) and RDP (Remote Desktop Protocol). While VNC is a cross-platform protocol, RDP is a Windows-only protocol. It supports most of the RDP-specific features as well.

KDE on a Mac OS X Snow Leopard

Mac OS X and Linux have more similarities than differences. They both have an open source kernel, both systems are built using GCC, both use default shell as Bash, both are UNIX systems (with Autotools compatibility), both use CUPS as their print back end... the list goes on. Thanks to Qt4/Mac, you can now add KDE to that list as well. Mac OS X is already a 'desktop environment' in the sense that it provides window management and application launching. What KDE aims to provide on Mac OS X are the KDE applications and the development libraries.

■ KDE application Marble Desktop Globe running on Mac OS X Snow Leopard

Perform the following steps to install KDE 4

1. Install Xcode SDK from **http://developer.apple.com**. You can also use the iPhone SDK to install the Xcode SDK. The Xcode SDK installs the mandatory applications for development, such as GCC, G++, LD etc .

2. Download the MacPorts installer from **http://distfiles.macports.org/MacPorts/MacPorts-1.9.2-10.6-SnowLeopard.dmg** and install it.

3. Update MacPorts with the latest port files:

```
$ sudo port -v selfupdate
```

4. Install KDE packages :

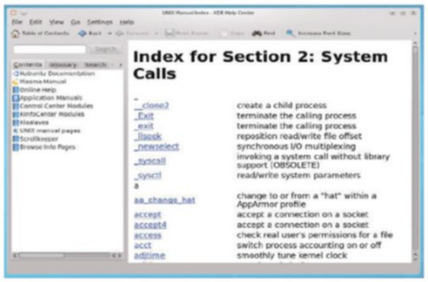
```
$ sudo port install kde*4*
```

This will install all KDE 4 packages, but you can also specify individual packages such as kdebase4, kdemultimedia4 etc, to install just the respective packages. To see a list of KDE 4 packages available, run the following command:

```
$ sudo port list kde*4*
```

5. Now wait. Since this is a source-based installation and everything will be compiled and then installed, expect this process to be little time-consuming. By the time the installation finishes, your system will have KDE 4 packages as well as all the KDE development libraries installed.
Pay attention to the following terminal messages which will be displayed post-installation.

```
############################################################
  # Don't forget that dbus needs to be started as the local
  # user (not with sudo) before any KDE programs will launch
  # To start it run the following command:
  # launchctl load /Library/LaunchAgents/org.freedesktop.dbus-session.plist
############################################################

#######################################################
  # Programs will not start until you run the command #
  # 'sudo chown -R $USER ~/Library/Preferences/KDE'  #
  # replacing $USER with your username, followed by  #
  # 'open /Applications/Macports/KDE4/kdeinit4.app'  #
  # in a Terminal.                  #
#######################################################
```



KRDC setting up RDP options

4. Sharing the desktop

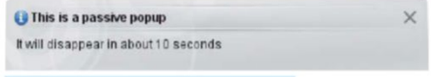
Sharing the desktop inside KDE is very easy. You can use Krfb, a server application that allows you to share your current session with a user on another machine, who can use a VNC client to view or even control the desktop. The username and password required to connect is automatically generated by the applications, can be used only once and expires in one hour. This provides an easy and secure way to share the desktop

5. Shell scripting the KDE way

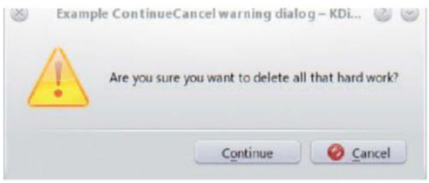
Shell scripting provides an easy way to automate tasks. It is also an indispensable tool for system administrators who use it for a range of tasks such as keeping system updates or creating default passwords for new users. Often these shell scripts are dull and only provide text-based interactions and output. However, you can use KDE Dialogs to provide a semi-graphical or completely graphical interface for interacting with shell scripts. You can make use of many common KDE UI elements such as text box, progress bar, message box, passive pop-up box etc. Above all, using these native UI elements does not require traditional programming and they can be easily called using shell scripts.

For example:

```
passivepopup dialog box
$ kdialog --title “This is a passive
popup” --passivepopup \ “It will
disappear in about 10 seconds” 10
```




```
--warningyesno message box
$ kdialog --title “Example
ContinueCancel warning dialog” \
--warningcontinuecancel “Are you sure
you want to delete all that \
hard work?”
```



6. Reading man and info pages

Man and info pages provide extensive documentation that comes pre-installed with most UNIX commands. You can use KDE Help Center to view man and info pages graphically. You can also do a full text search inside man and info pages along with other documentation installed in KDE Help Center.

7. Auto-starting applications

The applications to auto-start are indicated by .desktop files in the $KDEDIR/share/autostart directory. **You can also check for the personalised autostart directory using:**

```
$ kde4-config --userpath autostart
/home/kunal/Autosart
```

Auto-start of an application can be made available using the X-KDE-autostart-condition entry in the .desktop file. The ktip.desktop file (for KDE Tips), for instance, may contain

```
X-KDE-autostart-condition=ktiprc:
TipOfDay:RunOnStart:true
```

The above specifies an autostart condition where the ktiprc configuration file is checked for a RunOnStart entry in the [TipOfDay] section. If no such entry is found, true is assumed, which means that ktip will auto-started by default.

8. Configuring KDM for automatic login

If you are the only user on the system, it might annoy you to type your username and password to log in. Most of us would like to have an auto-login set up in this scenario. To configure automatic login, open KDE System Settings, then start the Login Screen module, which is present in the System Administration section. Open the Convenience Tab, check the Enable Auto-Login option and select a use for Auto-Login from the drop-down menu. You can also enable password-less logins by checking the Enable Password-Less Logins option. When you are happy with the changes, click Apply. You may be asked for further authentication to apply the changes.

KDE developer tricks

1. Developing for Unicode

Unicode allows you to mix different scripts and languages inside one document. It also makes it easy to interchange information between people with different locales. There are two unicode standards, namely UTF-8 and UTF-16. UTF-8 is a variable-width encoding, with each character represented by 1 to 4 bytes. UTF-16 signifies that every character is represented by the 16-bit value of its Unicode number. QString and QChar internally represents characters using UTF-16. **To convert between representations, you can use a QTextCodec. You can get a utf-8 and utf-16 using the following:**

```
QTextCodec Utf8Codec = QTextCodec::
codecForName(“utf-8”);
QTextCodec Utf16Codec = QTextCodec::
codecForName(“utf-16”);
```

You can read a Unicode text file using the QTextCodecs for UTF-8 data as follows:

```
QTextStream textStream;
QString line;
// UTF-16, if the file begins with a
Unicode mark
// the default is ok, otherwise:
// textStream.
setEncoding(QTextStream::Unicode);
line = textStream.readLine();
// UTF-8
textStream.setCodec(Utf8Codec);
line = textStream.readLine();
```

2. Accessing KDE services

The KService class is used to interact with KDE services: applications, plug-ins or other add-ons. You can use KDE services to query its name, find an associated icon, launch applications or load plug-ins. You can either create a KService object or request one by name. You can also get the KService object by querying.

The following example explains the KService object creation:

```
QString pathToDesktopFile =
KStandardDirs::locate(“services”,
“konqueror.desktop”);
KService service(pathToDesktopFile);
```

3. Make KDE understand more data using KIO slaves

A KIO slave is a library with a collection of all the supported file management functions. The KDE protocols that we talked about in the

Administrator section are actually backed by KIO slaves. This allows a KDE application to support a particular stream (or file) type by just using its respective KIO slave. You can also use KIO slave API to contain file management functions for a stream or file type.

4. Text-to-speech

Jovie is the KDE text-to-speech subsystem. It provides a plug-in based service that allows any KDE (or non-KDE) application to speak using the D-Bus interface. Applications can send text they wish to be spoken via D-Bus.

For example:

```
# Start Jovie (if not already
running)
jovie
# Send “Hello World” to KTTSD for
speaking in English.
qdbus org.kde.KSpeech /KSpeech say
“Hello World” 0
```

From the application, programmatically

```
org::kde::KSpeech* kspeech = new
org::kde::Kspeech(“org.kde.kttsd”,
“/KSpeech”,     QDbusConnection::
sessionBus());
kspeech->setApplicationName(“myappn
ame”);
kspeech->say(“Hello World”,
KSpeech::soPlainText);
```

5. Detecting and using hardware devices

To detect and use hardware devices, you can use Solid. Solid is the hardware device framework for KDE. You can use Solid to easily write applications with hardware interaction features. For example, you can use Solid to search for specific devices and find their device type.

Listing devices:

```
Solid::DeviceNotifier *notifier =
Solid::DeviceNotifier::instance();
foreach( Solid::Device device,
Solid::Device::allDevices() )
{
    //print the name of device
    kDebug() << device.udi();
}
```

Searching for a specific device:

```
Solid::Device::listFromType(Solid::
Capability::AudioHw,”real_specific
_parent” )
```

Querying device type:

```
QList<Solid::Device> list = Solid::
Device::listFromType(Solid::
DeviceInterface::Processor,
QString());
//take the first processor
Solid::Device device = list[0];
if(device.is<Solid::Processor>()
) kDebug() << “We’ve got a
processor!”;
else kDebug() << “Device is not a
processor.”;
Solid::Processor *processor =
device.as<Solid::Processor>();
kDebug() << “This processors maximum
speed is:” << processor->maxSpeed();
```

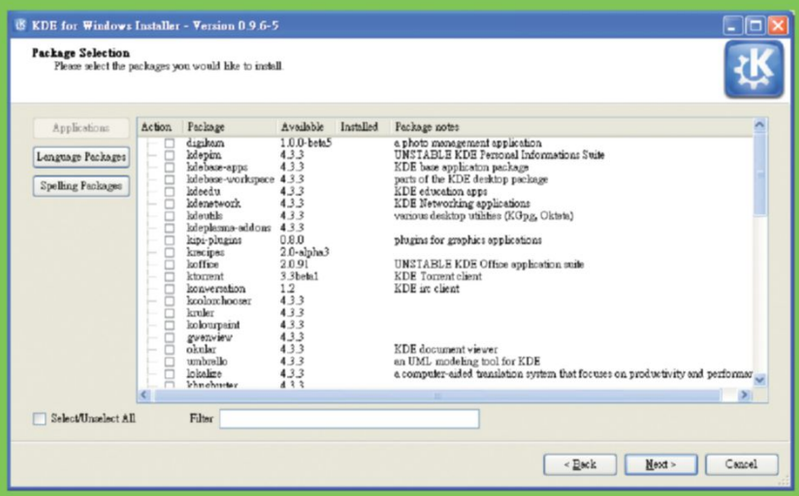
6. Developing for KDE in other languages

In addition to C++, KDE applications can also be developed in many other programming languages. Out of these programming languages, Python and Ruby are very popular within KDE. Most of the KDE APIs can be accessed using Ruby and Python. The Python-based implementation of KDE is called PyKDE4. It is developed by Riverbanking Computing Limited and is available from **www.riverbankcomputing.co.uk/software/pykde**. The Ruby-based implementation for KDE APIs is known as Korundum. Korundum is a RubyForge project and is available from **http://rubyforge.org/projects/korundum**.

Conclusion

KDE 4 has come a long way since it was founded in 1996. What started as a means of getting a nice user interface on top of UNIX operating systems has turned into a full-blown desktop environment. It not only caters to Linux, but also to other UNIX variants and Windows. No matter what system you use, the chances are that KDE can be installed there and you can reap the rewards it brings.

KDE on Microsoft Windows



UNIX and Windows are very different. They use different kernels, different development platforms... heck, even the makefiles for both platforms are different. Still, the folks at Nokia (formerly at Trolltech) managed to get 100% source compatibility with the release of Qt 4. This essentially means that all Qt 4 programs can be compile on Microsoft Windows without any hiccups. Built on Qt, most of the KDE libraries and applications are also available on Windows. This also means that developers can build Windows applications using KDE libraries. To install KDE on Windows, you can use an easy-to-use installation wizard available from **www.winkde.org/pub/kde/ports/win32/installer/kdewin-installer-gui-latest.exe**. Connect to the internet, run the installer and install the packages. This is not a source-based installation, so the installation will be considerably fast here. But there are still a few things to remember...

Install mode: KDE for Windows Installer offers two types of installation modes: 1) End User and 2) Package Manager. End User mode will only install the KDE applications and the runtime libraries. Developer Mode will also install the development libraries and the API documentation (where available).

Compiler mode: KDE primarily supports two types of compiler on Windows: MinGW (GCC port for Windows) and Microsoft Visual Studio or MSVC (the preferred compiler for the Windows environment). If you have a Windows-centric audience, you should select MSVC – otherwise, select MinGW as the compiler mode.

Everything you need to know about PHP – part 1

In this tutorial, Swayam Prakasha takes an introductory look at the popular general-purpose scripting language, PHP

```
[root@centos1 ~]# man php
PHP(1)                          Scripting Language                          PHP(1)

NAME
       php - PHP Command Line Interface ÁCLIâ

SYNOPSIS
       php [options] [ -f ] file [[--] args...]

       php [options] -r code [[--] args...]

       php [options] [-B code] -R code [-E code] [[--] args...]

       php [options] [-B code] -F file [-E code] [[--] args...]

       php [options] -- [ args...]

       php [options] -a

DESCRIPTION
       PHP  is  a widely-used general-purpose scripting language that is espe-
       cially suited for Web development and can be embedded into  HTML.  This
       is the command line interface that enables you to do the following:
```

A detailed look at PHP's command-line interface

```
[root@centos1 ~]# php --help
Usage: php [options] [-f] <file> [--] [args...]
       php [options] -r <code> [--] [args...]
       php [options] [-B <begin_code>] -R <code> [-E <end_code>] [--] [args...]
       php [options] [-B <begin_code>] -F <file> [-E <end_code>] [--] [args...]
       php [options] -- [args...]
       php [options] -a

  -a               Run as interactive shell
  -c <path>|<file> Look for php.ini file in this directory
  -n               No php.ini file will be used
  -d foo[=bar]     Define INI entry foo with value 'bar'
  -e               Generate extended information for debugger/profiler
  -f <file>        Parse and execute <file>.
  -h               This help
  -i               PHP information
  -l               Syntax check only (lint)
  -m               Show compiled in modules
  -r <code>        Run PHP <code> without using script tags <?..?>
  -B <begin_code>  Run PHP <begin_code> before processing input lines
  -R <code>        Run PHP <code> for every input line
  -F <file>        Parse and execute <file> for every input line
  -E <end_code>    Run PHP <end_code> after processing all input lines
```

How to use the PHP command-line interface

Resources

PHP is considered a general-purpose scripting language. Its original design was for web development and to produce dynamic webpages. One can treat PHP as an HTML embedded scripting language. Many of its syntaxes are similar to those available in C and Perl, but PHP provides a few unique features.

PHP has gained a lot of momentum because of its key features, such as:

- **A. It runs on various platforms**
- **B. It is compatible with all servers (such as IIS, Apache etc)**
- **C. It is freely available**

Before starting this tutorial on PHP, it is important that readers have a basic understanding and experience in the following:

- **A. HTML – You need to know the syntax and especially HTML Forms.**
- **B. Basic programming knowledge – This is not obligatory, but if you have any experience of any of the traditional programming languages, it will make your PHP learning much easier.**

PHP can be installed on both Windows and UNIX/Linux operating systems. The best way to install PHP on a UNIX/Linux environment is to go for a compile and configure process. Some of the prerequisite knowledge and software required for this are:

- **A. Fundamentals of UNIX scripting**
- **B. Exposure to Make utility and C compiler**
- **C. A web server**

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in information technology for several years. He can be reached at swayam.prakasha@gmail.com

Let us understand precisely how PHP works... When someone visits your PHP webpage, your web server processes the PHP code. It then sees which parts it needs to show to visitors (content and pictures) and hides the other part (such as file operations, maths calculations etc.). It then translates your PHP into HTML. Once the translation is completed, it sends the webpage to your visitor's web browser. PHP comes with a very useful command-line interface and it is worth to take a look at its man page, which we have captured it in the screenshot at the top of this page.

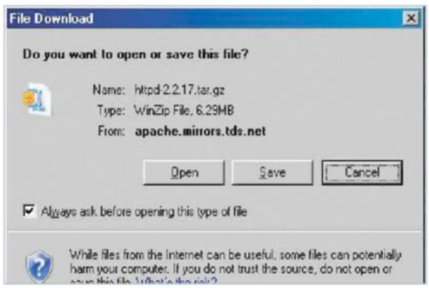
It is important to understand the various options that are available with the PHP command-line interface. The screenshot above takes a look at these options.

1. Installing a web server

As mentioned earlier, one needs to install a web server on your system so that you can run PHP scripts. As we all know, Apache is the most widely used web server. It is always good practice to install Apache first and then install PHP.

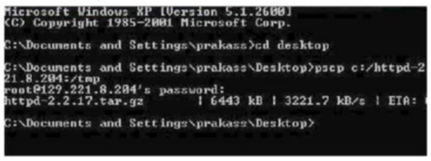
2. Download Apache source code

As always, make sure that you download the source code from a reliable source. Visit the Apache HTTP server download page and select a stable release. We have selected Apache release 2.2.17 for our practical purposes. Download the source code and save it on one of your drives on your system.



3. Transfer the source code on to your Linux box

One can use FTP (File Transfer Protocol) to transfer the code from your local drive onto your Linux system. It is not strange to see that on some systems, the FTP service will be disabled. In such cases, one can use PSCP to transfer the file securely.



■ Transfer the file using PSCP

4. Extract the source code

Extracting the code consists of uncompressing and untarring. This can be achieved by executing the following two commands.

```
[root@centos1 2.2]# guzip httpd-
2.2.17.tar.gz
[root@centos1 2.2]# tar -xvf httpd-
2.2.17.tar
```

Note that the above extraction creates a directory httpd-2.2.17 and it is good to have a look at the contents of this directory. This directory contains the source code and so you need to get into this directory for configuring and compiling the code.

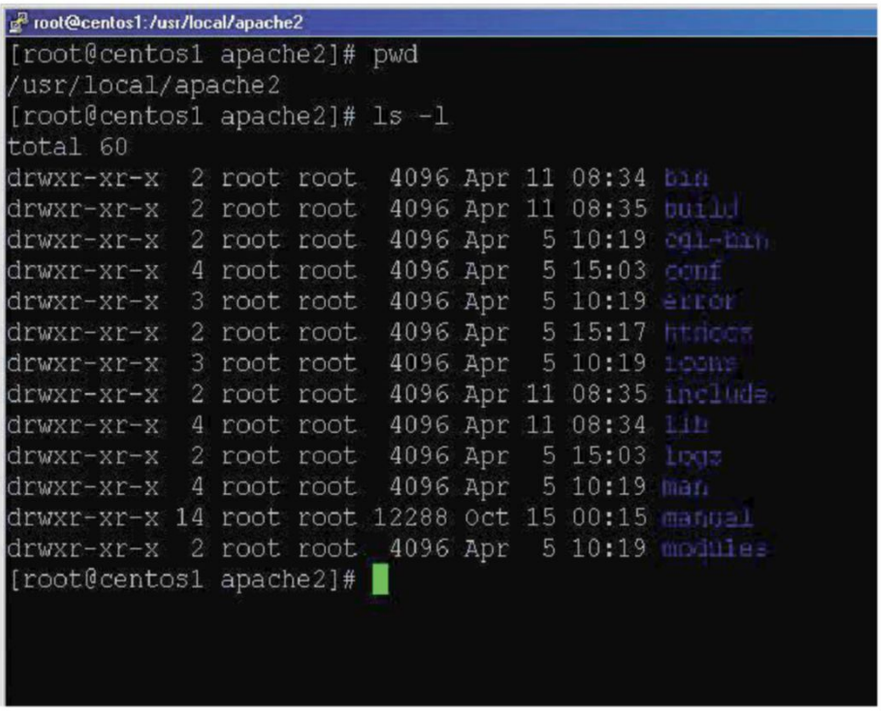


Fig 1 Configuring the source Various directories at /usr/local/apache2drive

5. Configuring the source

Once we have the source code, the next step is to configure the Apache source tree. This can be done by using the configure script. This script comes with many options. If you want to install Apache at a specific location, please make sure that you use the '--prefix' option. By default, Apache gets installed at /usr/local/apache2. There are various directories that are important to us at this location (**Fig 1**).

Another feature of interest here is that we can specify which features need to be included by enabling and disabling the modules. This will have a significant impact on the size of the binary.

6. Compilation and installation

Compilation and installation is done by executing the following commands.

```
[root@centos1 2.2]# make
[root@centos1 2.2]# make install
```

Please note that typically it requires root privileges to execute the above commands.

7. Customising the web server

Apache comes with a very useful configuration file, httpd.conf, and one can find this file at /usr/local/apache2/conf. If we need to customise Apache, then we need to edit this configuration file. This configuration file has various directives and their associated values. We can edit this file to change the value of a directive. And for this change to be effective, the server needs to be restarted.

8. Starting and stopping the web server

Apache provides a control script, called apachectl, and this script can be used to start and stop the server. Note that whenever we make any changes to the configuration file, the Apache server needs to be restarted. When the server is up, the contents of the file httpd.pid indicate the PID (process id) of the server.

9. Time to test our server now

Once you have successfully installed the web server, you can test it by firing a request using a browser. We have installed Apache on our Linux box (with an IP 129.221.8.208) and configured it to listen at the port 8080. As illustrated in the following screenshot (**Fig 2**, overleaf), our specific request displays the contents of index.html which is located at /usr/local/apache2/htdocs.

Fig 2 Testing the web server Test the web server after a successful installation

10. Debugging in a web server

Apache Web Server provides two log files – access_log and error_log – and they are pretty useful in debugging. For every successful request, access_log file will be populated; while an entry will appear in error_log when the request is not through.

11. Installing PHP on Linux

On a fully functional production server, in order to compile a newer version of PHP, we need to have the source library or other required header/files of your modules/packages to include their support. One needs to note that this can vary from a server to server. As mentioned earlier, the best way to install PHP is to go for the compile and configure option. We will explain this in the subsequent steps.

> **"The configure script performs system checks"**

```
[root@centos1 ~]# wget http://us3.php.net/distributions/php-5.3.5.tar.gz
--2011-04-08 09:42:35--  http://us3.php.net/distributions/php-5.3.5.tar.gz
Resolving us3.php.net... 192.41.42.26
Connecting to us3.php.net|192.41.42.26|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14105504 (13M) [application/x-gzip]
Saving to: `php-5.3.5.tar.gz.1'

100%[======================================>] 14,105,504   197K/s   in 2m 25s

2011-04-08 09:45:02 (94.9 KB/s) - `php-5.3.5.tar.gz.1' saved [14105504/14105504]

[root@centos1 ~]#
```

Fig 3 Download the source code Use Wget to download the code

```
[root@centos1 ~]# tail -f wget-log
Resolving us3.php.net... 192.41.42.26
Connecting to us3.php.net|192.41.42.26|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14105504 (13M) [application/x-gzip]
Saving to: `php-5.3.5.tar.gz'

100%[======================================>] 14,105,504   183K/s   in 71s

2011-04-08 08:28:50 (193 KB/s) - `php-5.3.5.tar.gz' saved [14105504/14105504]
```

Fig 4 Contents of wget-log file Make sure you've downloaded the complete source code

12. Download the PHP source code

Make sure that you download the PHP source code from a reliable source. One can use Wget (which is a non-interactive network downloader) for downloading purposes (**Fig 3**). We have taken 5.3.5 as the stable release of PHP.

13. Take a look at the log file

The download using Wget generates a log file, wget-log, in the same directory. It is always good practice to take a look at the contents of the log file (**Fig 4**) and make sure that we have the required source code in zipped format.

14. Extract the source code

To extract the PHP source code, execute the following two commands.

```
[root@centos1 ~]# gunzip php-5.3.5.tar.gz
[root@centos1 ~]# tar -xvf php-5.3.5.tar
```

```
[root@centos1 ~]# ls php-5.3.5.tar.gz
php-5.3.5.tar.gz
[root@centos1 ~]# gunzip php-5.3.5.tar.g
[root@centos1 ~]#
[root@centos1 ~]# ls php-5.3.5.tar
php-5.3.5.tar
[root@centos1 ~]#
```

Use gunzip to uncompress the code

15. Compile the source code and install PHP

After extracting the code, you need to run the configure script. This script performs various system checks. It also generates the required configuration files. Once we are through with a successful configuration, the next step is to go for compiling the source and installing PHP. This requires the execution of the following commands.

```
[root@centos1 2.2]# make
[root@centos1 2.2]# make install
```

16. Verify your PHP installation

We can verify our PHP installation by executing the following command.

```
[root@centos1 php-5.3.5]# php -v
```

A successful installation of PHP will display the following.

```
PHP 5.3.5(cli) (built: Nov 13 2009 11:24:03)
Copyright (c) 1997-2009 The PHP Group
Zend Engine v2.2.0, Copyright (c)
```

```
1998-2009 Zend Technologies
[root@centos1 php-5.3.5]#
```

17. Edit /etc/php.ini file

Edit the /etc/php.ini file and change the values of the below-mentioned parameter. Make sure that you uncomment these lines if they are commented.

```
short_open_tag = On
magic_quotes_gpc = On
register_globals = On
session.save_path = "/var/lib/php/
session"
```

18. Changes required in the Apache configuration file

Edit the Apache configuration file with the changes shown below.

```
LoadModule php5_module /usr/lib/
httpd/modules/libphp5.so
```

Next, search for AddType in the file and add the following lines after the last AddType statement. Add the line just before the closing </IfModule> for that section.

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-
source .phps
```

19. Restart the web server

As mentioned earlier, whenever we make changes to the configuration file, the Apache server needs to be restarted so that those changes will be effective. Execute the following command to restart the Apache server.

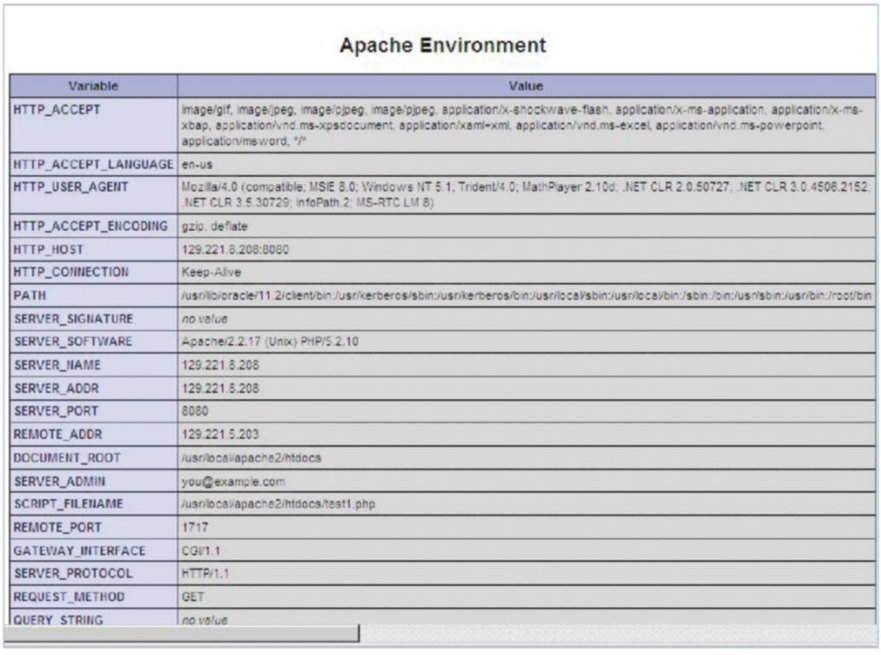
```
[root@centos1 bin]# ./apachectl
restart
```

It is always good to run a test page to make sure that the Apache server is running perfectly.

PHP Version 5.2.10

System	Linux centos1.centos.com 2.6.18-164.el5 #1 SMP Thu Sep 3 03:33:56 EDT 2009 i686
Build Date	Nov 13 2009 11:25:53
Configure Command	'./configure' '--build=i686-redhat-linux-gnu' '--host=i686-redhat-linux-gnu' '--target=386-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix' localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=./config.cache' '--with-libdir=lib' '--with-c dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg- enable-magic-quotes' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '--enable-track-vars' '--enable-trans-sid' '--enable-y -dbx' '--enable-dio' '--without-mime-magic' '--without-sqlite' '--with-libxml-dir=/usr' '--with-xml' '--with-system-tzdata' '--with-apxs2=/usr/sbin/apxs' '--with json'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
additional .ini files parsed	/etc/php.d/dbase.ini, /etc/php.d/gd.ini, /etc/php.d/json.ini, /etc/php.d/ldap.ini, /etc/php.d/mbstring.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/o
PHP API	20041225
PHP Extension	20060613
Zend	220060519

Fig 5 **Execution of first script** A part of the display of our first script



Apache Environment

Variable	Value
HTTP_ACCEPT	image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, application/x-ms-application, application/x-ms-xbap, application/vnd.ms-xpsdocument, application/xaml+xml, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
HTTP_ACCEPT_LANGUAGE	en-us
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; MathPlayer 2.10d; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.2; MS-RTC LM 8)
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_HOST	129.221.8.208:8080
HTTP_CONNECTION	Keep-Alive
PATH	/usr/lib/oracle/11.2/client/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
SERVER_SIGNATURE	no value
SERVER_SOFTWARE	Apache/2.2.17 (Unix) PHP/5.2.10
SERVER_NAME	129.221.8.208
SERVER_ADDR	129.221.8.208
SERVER_PORT	8080
REMOTE_ADDR	129.221.5.203
DOCUMENT_ROOT	/usr/local/apache2/htdocs
SERVER_ADMIN	you@example.com
SCRIPT_FILENAME	/usr/local/apache2/htdocs/test1.php
REMOTE_PORT	1717
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	no value

Fig 6 **Executing the PHP script** Apache environmental details of our first script

20. Our first PHP script

Here is our first PHP script. There are many similarities to other programming languages (such as Perl and C). The only addition is that the entire PHP code should be enclosed within a tag.

```
<?php
 phpinfo();
?>
```

In this script, we've used a PHP built-in function. Place this script under DocumentRoot (by default, this location will be /usr/local/apache2/htdocs). Make sure that the script has the required execute permissions. Also, make sure that you save the PHP script with a .php extension (we saved it as test1.php).

21. Executing the PHP script

We can execute our script by firing a request through a browser (**Fig 5** & **6**). This request can have three parts: IP address of our Linux machine, the port configured for the Apache web server and name of the PHP script. Thus the URL to execute our first script looks like this:

```
http://129.221.8.208:8080/test1.php
```

In this tutorial, we followed a step-by-step process to install an Apache web server. We also understood how one can install PHP and execute a simple script. In future issues of **LUD**, we will examine more in-depth concepts in PHP.

Everything you need to know about PHP – part 2

Swayam Prakasha looks at some more features of the world-renowned general-purpose scripting language, PHP

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in information technology for several years. He can be reached at swayam.prakasha@gmail.com

Resources

In the first part of this article (which appeared in issue 101), we took a look at some of the important aspects as far as installation and configuration are concerned. With that, we basically did some groundwork so that we can go ahead with more aspects of this interesting scripting language – PHP. We noticed that the most preferred way to install PHP on a UNIX/Linux environment is to go for a compile-and-configure process with some prerequisites such as UNIX scripting, web server and makefile etc. It is essential to install a web server on your system so that one can run PHP scripts. By taking Apache as our standard web server, we understood the various steps involved in installing this web server. Typically, the installation consists of the following important steps...

a. Downloading and extracting the source code
b. Configuring the source code
c. Compilation and installation
d. Customising the web server (by modifying the contents of the configuration file – httpd.conf)
e. Use the control script to start, stop and restart operations

Here, we will take a look at some of the intermediate concepts of PHP. Our focus will mainly be on various topics such as strings, operators, conditional statements, arrays and looping constructs.

In part one of our guide to PHP, we looked at the procedure for installing PHP on a Linux machine. We also examined the various configuration changes required – especially to /etc/php.ini and Apache configuration file httpd.conf. Once we had completed the installation of web server and PHP, we executed our first PHP script and observed its output. In that script, we noticed that PHP scripts are always enclosed in between two PHP tags, as shown here:

```
<?
PHP code comes here
?>
```

In this, the second tutorial in the series, we will take a closer look at some of the important concepts of PHP, such as variables, echo, strings, operators, conditional statements (for example, if....else), switch, arrays and various looping constructs.

Note that we need to follow the steps shown below so that PHP scripts can be executed.

a. Save the file with a .php extension
b. Place the file in a PHP-enabled server
c. Use a web browser to load this file

1. Time to start with a simple PHP script

Let us write our first PHP example. **The following is a simplest PHP and HTML page that we can create.**

```
<html>
<head>
<title> This is Our First PHP Example</
title>
</head>
<body>
<?php
echo “Welcome to LU&D!”;
?>
</body>
</html>
```

After giving the required execute permissions to the above PHP script, we can execute it by firing a request through a browser:

```
http://129.221.8.208:8080/test2.php
```

We can see from this that 'Welcome to LU&D!' will be displayed.

2. Terminating the PHP code

If we take a careful look at our previous PHP code, we can see there is a semicolon at the end of the code. It is important to note that the semicolon always indicates the end of a PHP statement. **Take a look at the following example and analyse its output.**

```
<html>
<head>
<title> This is Our First PHP Example</
title>
</head>
<body>
<?php
echo “Welcome to LU&D!”;
echo “Welcome to PHP tutorial!”;
?>
</body>
</html>
```

3. Variables in PHP

As we know, a variable allows us to store a value. Defining a variable in PHP is similar to the way we define it in other scripting languages. **Take a look at the following sample to understand how we can define variables in PHP.**

Hello, How are you? 4

■ Understanding variables in PHP

```
<?php
$my_string = “Hello, How are you?”;
$my_num = 4;
?>
```

Please note that there is no need to declare the variables in PHP, but we need to follow certain defined rules while naming the variables.

4. Outputting text to a web server

The PHP command ‘echo’ is widely used to output text to a web server. In our first PHP example, we have used echo to display a string. It is important to note here that echo is not a function, but a language construct. **Take a look at the following example and see its output.**

```
<?php
$my_string = “Here we are!”;
echo $my_string;
echo “<h5>Good to start learning PHP!</
h5>”;
?>
```

We need to understand here that since the text will be sent to the user in the form of a webpage, we need to make sure that we use proper HTML syntax.

One needs to be careful when strings with quotes are used. It is always required to ‘escape’ quotes that are within a string.

5. How to echo variables and text strings

Echoing variables is fairly simple in PHP. Look at the following section of a PHP code to understand this.

```
<?php
$statement = “Hello, My favourite
number is:”;
$number = 2011;
echo $statement;
echo $number;
?>
```

When this gets executed, it displays ‘Hello, My favourite number is:2011’.

PHP allows you to place variables inside a string that begins and ends with double quotes. When we do this, PHP basically takes the value of that variable and uses it in the string.

Hello, My favourite number is:2011

■ Reading the value of a variable inside a string

6. Strings in PHP

Once you create a string, you can directly use it in a function or it can be stored in a variable. As expected, it is always good to save your strings in variables if you plan to use them more than once. Double quotes are normally used for forming strings. And as seen earlier, a backslash is used to escape a character. Note that if you try to escape a character that doesn’t need to be, then the backslash will also show up when you output that string.

7. String creation using heredoc

PHP provides another very useful tool for creating strings and it is known as heredoc. This tool allows the programmer to create multiple-line strings without using any quotations. A programmer needs to be very careful when he uses this tool for creating strings in his PHP code. Let us understand this with an example.

Our PHP code is given below.

```
$my_string = <<<SAMPLE
Linux User and Developer
PHP Learning
Unlock your potential!
SAMPLE;

echo $my_string;
```

It is important to note that the output will not spread across multiple lines as we have not used any
 tags.

Linux User and Developer PHP Learning Unlock your potential

■ Creating strings using the heredoc tool

8. Operators in PHP

As we all know, in any programming language, operators are typically used to perform various operations on variables. And operators in PHP are no exception in this regard. **Operators in PHP can be classified into the following categories:**

a. **Assignment operators**
b. **Arithmetic operators**
c. **Comparison operators**
d. **String operators**

9. Assignment operators

One can use assignment operators to set a variable equal to a value, or set a variable to another variable's value. It is natural to use these operators in conjunction with arithmetic operators.

Here are a couple of examples of assignment operators:

```
$Var1 = 10;
$Var2 = $Var1;
```

10. Arithmetic operators

PHP provides various arithmetic operators and they are +, -, *, / and %. **We have used a few of them in our example below.**

```
$add_result = 3 + 6;
$div_result = 17 / 4;
$mod_result = 9 % 2;
echo "Performing addition: 3 + 6 =
".$add_result."<br />";
echo "Performing division: 17 / 4 =
".$div_result."<br />";
echo "Performing modulus: 9 % 2 = " .
$mod_result. ".
```

11. Comparison operators

Comparison operators are used to check the relationship between variables (and/or values). These operators are used inside a conditional statement and will get evaluated to true or false. Some of the commonly used comparison operators are shown in the table above.

Operator	Meaning	Example	Result
==	Equal To	$x==$y	False
!=	Not Equal To	$x !=$y	True
<	Less Than	$x<$y	True
>	Greater Than	$x >$y	False
<=	Less Than or Equal To	$x<=$y	True
>=	Greater Than or Equal To	$x >=$y	False

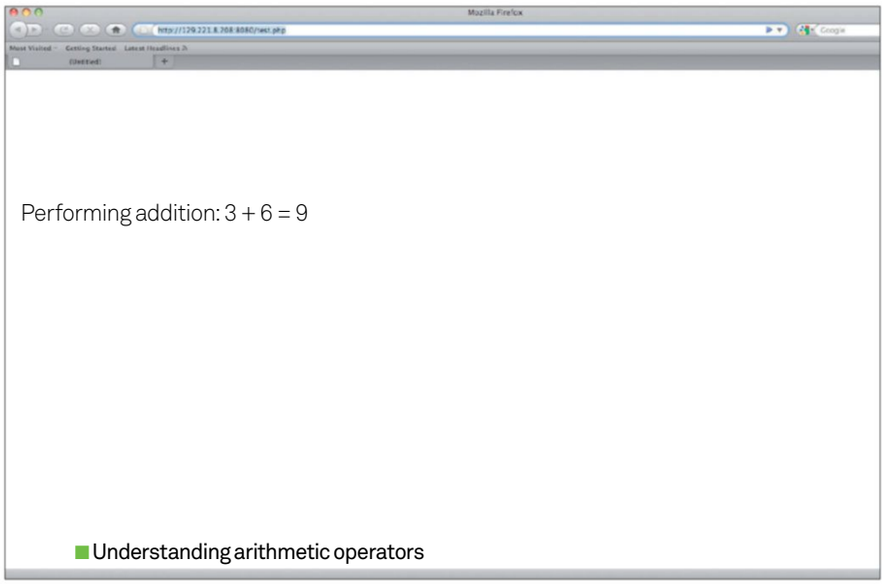
Some common comparison operators

12. String operators

The operator '.' (ie full stop/period) is used to concetenate two or more strings. **Take a look at the following sample code to learn more about this operator.**

```
$first_string = "Hello";
$second_string = " Good Morning";
$resultant_string = $first_string .
$second_string;
echo $resultant_string . "!!";
```

The above code displays 'Hello Good Morning!!' as its output.



Understanding arithmetic operators

13. Increment and decrement operators

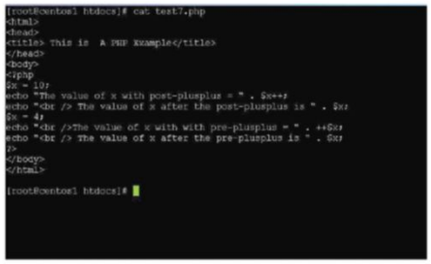
It is common to combine arithmetic and assignment operators. **An example for this is a count. Normally, to increment a count, a programmer will have:**

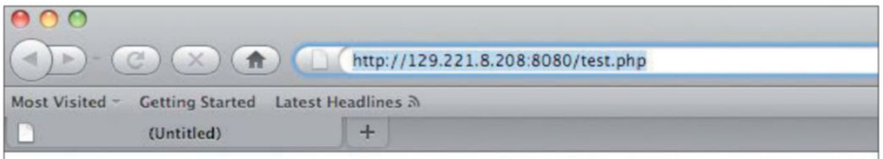
```
$count = $count + 1;
```

And it is not surprising to see another format – $count +=1 – that performs the same operation. Thus combinations of arithmetic and assignment operators achieve the same task. On the downside, there will be a reduction in the code readability when developers decide to use such combinations.

PHP also supports pre/post-increment and pre/post-decrement operators. To add 1 to a variable (ie to increment), we use the '++' operator; and '--' for a decrement operation. The advantage with these operators is that a developer can specify whether he wants to increment (or decrement) before a line of code is executed or after the execution of a line of code.

You can see from the screenshot below that the value of $Var++ is not reflected in the echoed text because the variable is not incremented until after the line of code is executed. However, with the pre-increment, ++$Var, the variable reflected the addition immediately.






```
The value of x with post - plusplus =10
The value of x after the post - plusplus = 11
The value of x with with pre - plusplus = 5
The value of x after the pre - plusplus = 5
```

■ Displaying the script output

14. PHP include command

PHP provides a very useful command – include – and one can use it to save a considerable amount of time. This command is useful whenever we need to do a task several times. Typically, the include command takes a filename and simply inserts that file's contents into the script that issued the include command.

Here is the syntax of this command:

```
<?php include("tasks.php");
?>
```

15. If...else conditional statement

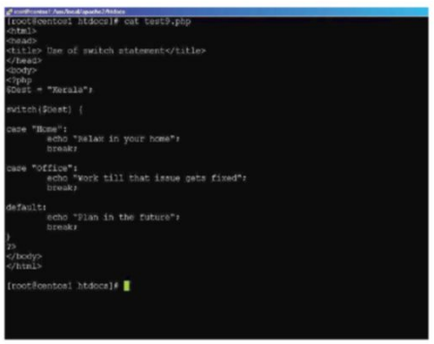
The PHP 'if' statement is pretty similar to the use of if in other programming languages. Whenever you want to make a decision based on whether something is true and be sure that you take the appropriate action, you are going to use an if/then relationship. **A simple example illustrating these constructs is given below.**

```
<html>
<head>
<title> Use of conditional constructs</
title>
</head>
<body>
<?php
$var = 10;

if ($var == 10)
        echo "No change in the value of
variable";
else
        echo "Value of variable is not
10";
?>
</body>
</html>
```

16. PHP switch statement

When a programmer needs to check for multiple conditions at once, the switch statement becomes pretty handy. Basically, the switch statement takes a single variable as input and checks its value against different cases that appear in the switch statement.

We have illustrated the use of the switch statement in the screenshot below.



■ A sample script to demonstrate the use of the switch statement

17. PHP array

As we all know, an array is a data structure that stores one or more values in a single value. We use key/value pairs to represent the elements of an array. **An element of an array has the following format.**

```
$array[key] = value;
```

It is important to note here that an array's keys start from 0 and not from 1.

18. While loop

Normally, programmers will use loops to do something over and over again until a specific task is completed. The 'while' loop is used to do a task over and over as long as the specified conditional statement is true. **The syntax of the while loop is given below.**

```
while ( conditional statement is true){
   //do this code;
}
```

We can note here that if the conditional statement is true, the code within the while loop will be executed. If the conditional statement becomes false, then the code following the while loop is executed as normal.

19. For loop

The 'for' loop has some more code added to it. A for loop can be considered to consist of four steps...

a. Set a variable (known as counter) and initialize it to some value
b. Check if the conditional statement is true
c. Execute the code within the loop
d. Increment the counter at the end of each iteration through the loop

In this second part of our PHP tutorial series, we have taken a look at some of the intermediate concepts of PHP. Concepts such as strings, operators, conditional statements and looping constructs are very useful for the day-to-day operations of system administrators.

In the forthcoming issues of **Linux User & Developer**, we will explain and demonstrate some of the more advanced concepts in PHP. So make sure you don't miss an issue –the best way is to take out a subscription, of course!

Everything you need to know about PHP – part 3

In the final part of his PHP series, Swayam Prakasha looks at some of the advanced features of this general-purpose scripting language

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in information technology for several years. He can be reached at swayam.prakasha@gmail.com

We hope that you enjoyed the earlier two parts of this tutorial series about PHP (they appeared in issues 101 and 102 of Linux User & Developer). In the first part, we took a look at some of the important aspects as far as installation and configuration are concerned. Therein, we noticed that the best way to install PHP on UNIX / Linux environment is to go for a compile-and-configure process with some prerequisites such as UNIX scripting, web server and makefile etc. We also learnt that it is essential to install a web server on your system so that one can run PHP scripts. In the second part, we focused on some further concepts, such as strings, operators, conditional statements, arrays and looping constructs. We were able to demonstrate their use with sample PHP scripts.

In part 1 of our PHP, we took a look at the procedure for installing PHP on a Linux machine. We also understood the various configuration changes required – especially to /etc/php.ini and Apache configuration file httpd.conf. Once we had successfully installed our web server and PHP, we executed our first PHP script and observed its output. In that script, we noticed that PHP scripts are always enclosed in between two PHP tags, as shown below.

```
<?
PHP code comes here
?>
```

In this tutorial, we will take a closer look at some of the advanced concepts in PHP and our focus will be on files and strings in PHP.

As we all know, manipulating files is a very basic necessity for every developer. Fortunately, PHP offers a wide variety of tools and they can be used for the creating, editing and uploading of the files.

It is important to note here that we need to follow the following steps so that PHP scripts can be successfully executed:

a. Save the file with a .php extension
b. Place the file in a PHP-enabled server
c. Use a web browser to load this file

```
root@centos1:/usr/local/apache2/htdocs
[root@centos1 htdocs]# pwd
/usr/local/apache2/htdocs
[root@centos1 htdocs]# cat test_file_1.php
<html>
<head>
<title> This is Our First File Manipulation Example</title>
</head>
<body>
<?php
$FileName = "test.txt";
$FileHandle=fopen($FileName,'w') or die("can't open file);
fclose($FileHandle);
?>
</body>
</html>

[root@centos1 htdocs]#
```

1. Creating a file

In PHP, the function 'fopen' is used to open files. It will also create a file if the file specified does not exist. We need to pass two important pieces of information to the fopen function:

a. The name of the file that we want to open
b. What we are planning to do with the file (read, write etc)

We illustrate this concept in the following script:

```
$FileName = "test.txt";
$FileHandle = fopen($FileName, 'w') or die("can't open file");
fclose($FileHandle);
```

After executing this PHP script, we can see the creation of the specified file and it will be placed in the same directory where our PHP code resides.

After giving the required execute permissions to the above PHP script, we can execute it by firing a request through a browser:

http://129.221.8.208:8080/test_file_1.php

2. File open

As mentioned earlier, the function fopen can be used to open an existing file. When we are opening a file, PHP expects us to specify our intentions. A file can be opened for 'read-only' mode, 'write-only' mode and 'append' mode. A file pointer is PHP's way of remembering its location in a file. When you open a file for reading, the file pointer begins at the start of the file.

If we carefully note the script in step 1, we have used the 'w' option – meaning that we want to open the file in 'write-only' mode.

3. Closing a file

It is important to note that the server will close all files after PHP code finishes execution. But it is always a good programming practice to close all files once you are done. In PHP, the function 'fclose' can be used to close a file. In the earlier example, we used fclose to close the file by passing the file handler as a parameter. Once we close a file, we will not be able to read, write or append to that file.

4. Writing into a file

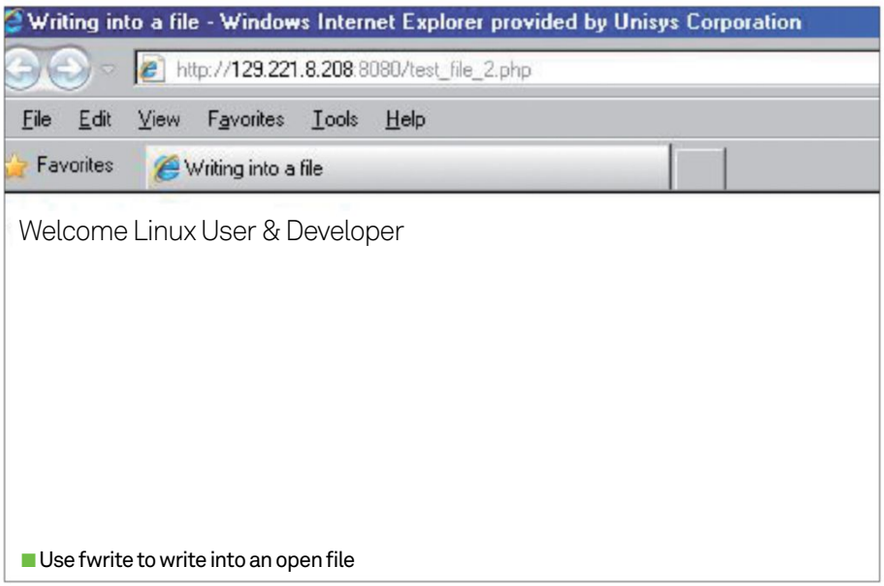
Now we know how to open and close a file, let us turn our attention to one aspect of file manipulation: writing. The function 'fwrite' can be used to write into a file. This function takes two parameters: the first one is the file handler and the second is the data string that is to be written. **Let us take a look at a sample script that explains this concept:**

```
$myFile = "test.txt";
$fh = fopen($myFile, 'w') or die("can't
open file");
$string = "Welcome\n";
fwrite($fh, $string);
$string = "Linux User & Developer\n";
fwrite($fh, $string);
fclose($fh);
```

After executing the above PHP script, the contents of the test file will be:

Welcome
Linux User & Developer

"PHP has a wide variety of tools to manipulate files"



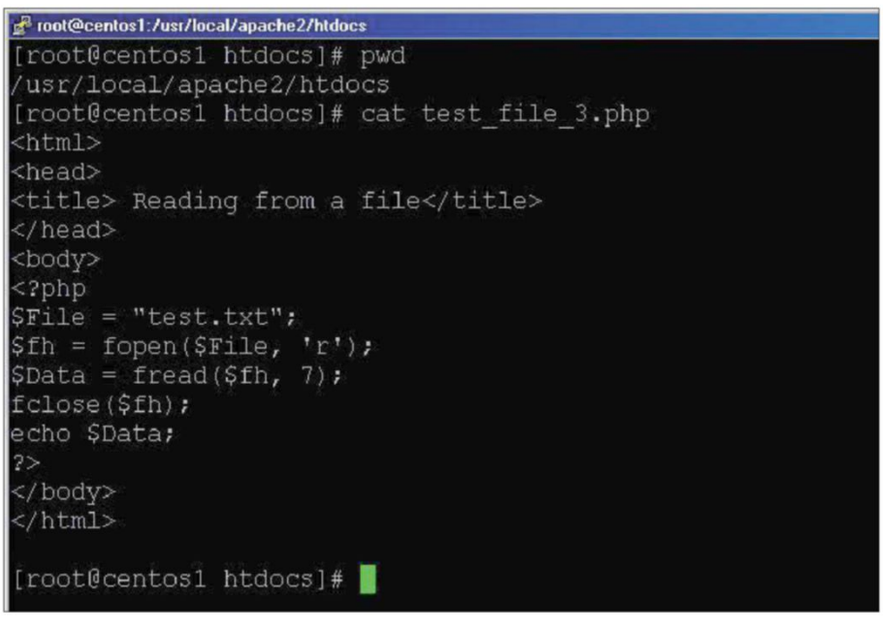
Use fwrite to write into an open file

5. Reading from a file

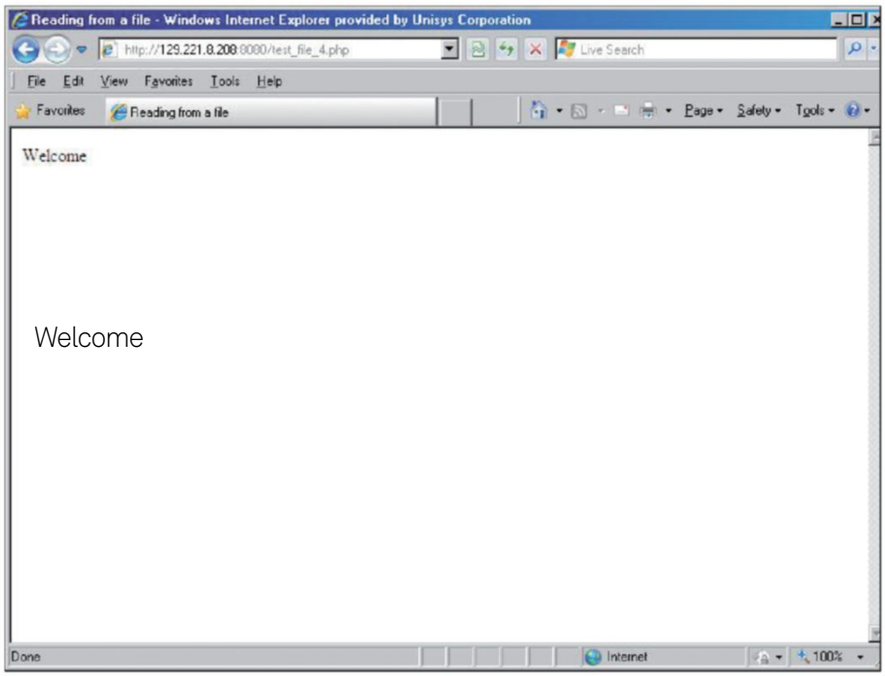
The function 'fread' is used for getting data out of a file. We need to pass two parameters to this function: the file handler and an integer value that specifies how much data (in bytes) it is supposed to read. Let us use a PHP script to read the contents of the file test.txt (that we have already opened in the previous example)

```
$File = "test.txt";
$fh = fopen($File, 'r');
$Data = fread($fh, 7);
fclose($fh);
echo $Data;
```

Here we try to read 7 bytes from our file. The last echo statement displays the string retrieved.



Read the contents of a file



Use the fgets function to read a file

6. Read a file using fgets

PHP allows you to read a line of data with a function: 'fgets'. If you had separated your data with new lines then you could read in one segment of data at a time with the fgets function. **Let us take a look at an example:**

```
$File = "test.txt";
$fh = fopen($File, 'r');
$Data = fgets($fh);
fclose($fh);
echo $Data;
```

This function is designed in such a way that it searches for the first occurrence of '\n', the new-line character.

7. Deleting a file

In PHP, you can delete files by using a function called unlink. Before you can delete a file, you need to make sure that it is not open in your program. You can use the fclose function to close down an open file and then delete it. PHP code to delete a file is **given below:**

```
$File = "test.txt";
unlink($File);
```

8. File append

If you are interested in adding data to the existing data in a file, then the file needs to be opened in the append mode. This can be done by passing 'a' as the second parameter to our fopen function call. This append functionality finds extensive use in web server logs. Web server logs are typically used to document events that occur over a period of time, rather than all at once.

"File manipulation is an interesting topic and PHP provides various means by which one can perform various operations on files"

9. Truncating a file

Sometimes we may need to truncate files that contain data that will only be used for a short time, before needing to be replaced. In PHP, when we open a file for writing (by passing w as the second parameter to an fopen call), it completely wipes all data from that file.

10. A look at magic quotes

Let us turn our focus to another important feature in PHP, called magic quotes. This feature will help to prevent new programmers from writing bad form-processing code. First, you need to see if magic quotes have been enabled on your server. **The following script will help us in this:**

```
if(get_magic_quotes_gpc())
   echo "Magic quotes are enabled";
else
   echo "Magic quotes are disabled";
```

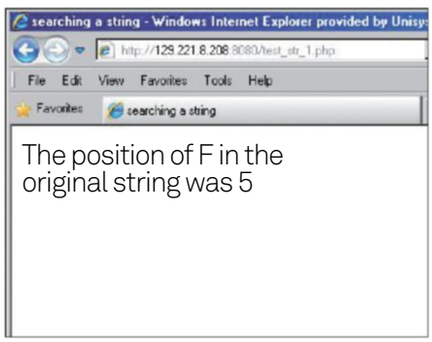
Magic quotes are designed to automatically escape risky form data that might be used for SQL injection with a backslash, '\'.

11. String manipulations – search a string

PHP provides various ways by which one will be able to manipulate strings. A function – strpos – is used for searching a string. This function takes two parameters and will return either true or false depending on the search result. **We will demonstrate this concept with an example:**

```
$String = "ABCDEFGHIJKLMN";
$To_Search = strpos($String, "F");
echo "The position of F in the original
string was $To_Search";
```

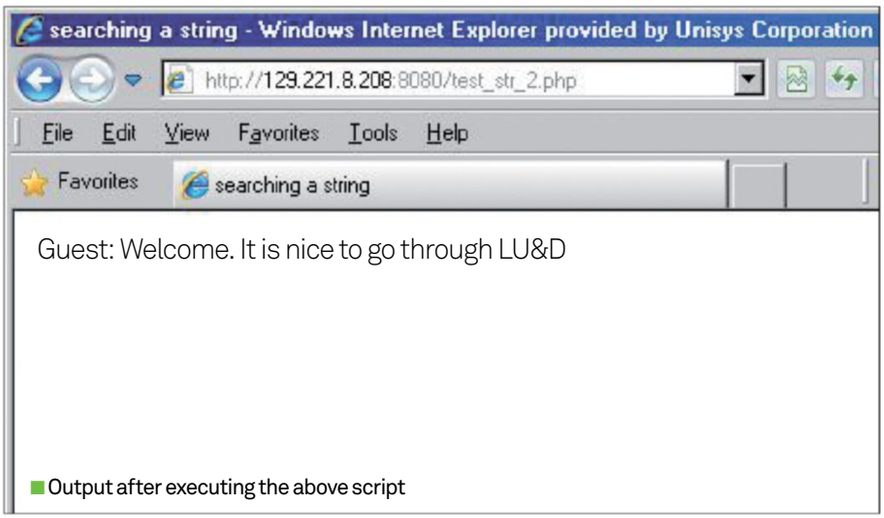
In PHP, since the counting starts from 0, the echo statement will display 5 as the position of F.



Understanding the string search function

```
root@centos1:/usr/local/apache2/htdocs
[root@centos1 htdocs]# cat test_str_2.php
<html>
<head>
<title> searching a string</title>
</head>
<body>
<?php
$String = "Welcome. It is nice to go through replaceme!";
$Alt_Str = str_replace("replaceme", "LU&D", $String);
echo "Guest: ". $Alt_Str . "<br />";
?>
</body>
</html>

[root@centos1 htdocs]#
```

A script to understand string replace functionality



Output after executing the above script

12. String manipulations – replace parts of a PHP string

Sometimes we may need to replace parts of a string. PHP provides a very useful function, 'str_replace' for this. **Its synax is as follows:**

```
str_replace(search, replace, originalstring)
```

We take a look at the function in this script:

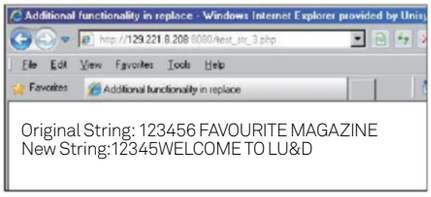
```
$String = "Welcome. It is nice to go through replaceme!";
$Alt_Str = str_replace("replaceme", "LU&D", $String);
echo "Guest: ". $Alt_Str . "<br />";
```

13. Additional functionalities of the replace function

PHP provides another function – substr_replace – that provides additional functionalities. This function uses starting points and lengths to replace parts of strings.

The following example explains this function in detail:

```
$String_Original = "123456 FAVORITE MAGAZINE";
$New_String = substr_replace($String_Original, "WELCOME TO LU&D", 5);
echo "Original String: $String_Original <br />";
echo "New String : $New_String <br />";
```

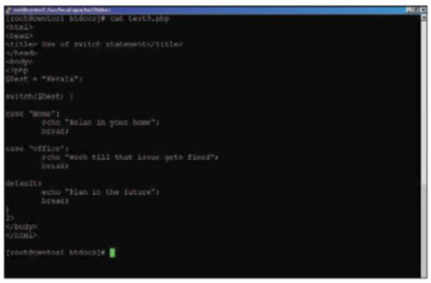


14. Capitalisation functions

There are three capitalisation-related functions in PHP. **They are:**

1. strtoupper
2. strtolower
3. ucwords

The first function takes a string and converts it into upper case. The second function takes a string as an input and outputs a lower-case version. The third function – ucwords – is used to capitalise the first letter of every word.



15. Explode function in PHP

The 'explode' function takes a string as an input and blows it into smaller pieces (based on a delimiter). It takes two parameters: the first is a delimiter and the second is the string. **We illustrate this function in a sample script:**

```
$Mobile_Number = "99804-05220";
$Mobile_Chunks = explode("-", $Mobile_Number);
echo "First part of Mobile number = $Mobile_Chunks[0]<br />";
echo "Second part = $Mobile_Chunks[1]<br />";
```

As can be seen from this example, the explode function returns an array of string pieces.

There is a converse function to explode and it is called implode. It can be considered as an 'undo' function of explode.

In this final part of our PHP tutorial, we looked at two important aspects: file manipulation and string manipulation. System administrators use these concepts widely in their day-to-day operations. Now it is time for you to go and work on these topics.

Open source and the law

Installing Linux may be free of charge but it's not free of liability...

As countries around the world now drag themselves out of recession, CIOs and their boards are looking once more at their IT needs. In many cases the 'pause period' that has been imposed over the last few years' worth of IT upgrades makes their impending roll-outs all the more important.

At the same time, the suppliers of software and services have been eagerly waiting the upgrade cycle to begin in earnest and have created what appears to be very compelling propositions for their customers. For many of these propositions, the devil is in the detail and that detail is in the contractual terms that accompany the software and services. This article considers the legal approaches that open source distros use to get paid for something that at first looks like a free lunch.

"Can a customer install RHEL but then use another Linux supplier to support it?"

Shareholder obligations

Before delving into some of the contractual niceties, it is worthwhile being reminded of some basic business rationales. All software and IT services companies listed on the stock market – from Red Hat to Microsoft and from Salesforce.com to Oracle – have financial obligations to their shareholders, who are keenly interested in money generated from return on capital investment. These companies have a duty to maximise this return to their investors. Even when software companies provide software or services for free, somewhere there must be a potential line in a profit and loss statement to show how directly or indirectly that free provision may turn into cash.

The business model of a Red Hat or a Novell SUSE would fail if the firm invested all its time in improving and distributing Linux but no customer took on its paid-for services – it needs to recoup its investment by ensuring that the customer takes both the software and the paid services on offer.

What makes the business proposition for a company like Red Hat particularly difficult is that it is providing its version of Linux software for free, while other close derivative versions of Linux are available from alternative suppliers such as CentOS, Scientific Linux and Oracle. Red Hat (and indeed Oracle) makes money on the support and services it provides to customers that are related to the so-called 'free' software. Indeed, on 22 September 2010, Red Hat reported that in the previous six-month period, its customers paid over US$400m for what it describes as, "Subscriptions, training and services."

So what makes a service?

Services often include technical support, bug fixes and software updates. So Red Hat, understandably, has sat down with its developer teams and lawyers to bond together a number of building blocks to try to tie in its customers to ensure that there is some commercial or legal obligation to pay Red Hat for services relating to Red Hat's version of Linux. Seasoned CIOs will see that these commercial and legal tricks are similar to those used by proprietary software companies over the years to tie their customers into buying services from them too.

Red Hat's marketing message is this: once a customer has obtained Red Hat Enterprise Linux (RHEL), the customer should also pay Red Hat for support. Sun Microsystems (under Jonathan Schwartz and even more so under King Larry) has always said exactly the same thing. A letter from Red Hat to a customer in January (pictured overleaf) shows this messaging. In its June newsletter to its partners, Red Hat went further; it went so far as to suggest that customers must de-install RHEL software from servers with expired subscriptions.

The big Linux legal question...

This article does not dwell on whether or not Red Hat does a good job in supporting Linux. What this article does test is whether anything legally prevents a customer from installing RHEL but then using another Linux supplier to support it.

Red Hat appears to use two approaches to discourage customers from receiving RHEL support from a third-party supplier. First, on commercial grounds it sets out in its FAQs (**www.redhat.com/rhel/renew/faqs/#6**) – and contract – the five rights that the customer would lose by appointing a different support provider. We analyse these below to discover that only one is actually relevant when moving to another supplier (rather than taking the support in-house, often referred to as 'self-supported'). Second, Red Hat has as recently as January this year threatened existing users that the likes of pre-acquisition (by

Why some companies demand commercial licences

While the technology being discussed here is still referred to as open source and in many cases FOSS (ie free and open source), the reality is that some corporations will automatically opt for commercial licence support. So why is this the case?

- Open source allergies: these can develop because the company runs a restricted set of mission-critical applications (possibly defence/military or financial or medical) where there is zero option for error correction.
- The company can not be involved in injunctions arising from errors caused by misfiring software that could usually be fixed in the normal course of business.
- There are too many warranties and indemnities associated with the company's business to engage any variables that are in any way unknown.
- There is an overriding need to prevent end users from accessing the code.
- There is a dominating need for support services from the start of the project to establish the company's IT stack.
- The business has combined its open source software with proprietary products that prohibit the use of LGPL or other licensing terms.
- The business is impelled to use commercially licensed open source software by investors who back the business's daily operation.

Attachmate) Novell was "not entitled by Red Hat to provide the Red Hat Enterprise Linux operating system, support, maintenance, version updates, bug fixes, patches or any other Red Hat certified software package or service".

We analyse this also to conclude that in relation to Linux software (as opposed to any Red Hat proprietary code), this is not sustainable.

Honour among software access thieves

Given that much of the software provided by Red Hat is under the GPL and is readily available, not having 'software access' to those elements seems no hardship since a customer's new support company will doubtless provide those updates. Certification certainly is something of value but given that the certification relates to GPL Linux, there is no valid legal restriction that Red Hat could impose on a new supplier to prohibit it from installing the same versions of the GPL code. It does, however, seem odd that certification of 'code and kit' could be lost just because a customer chooses another support provider.

A supplier's legal representations made at one point in time to a customer usually remain valid and do not often expire merely because a new supplier is appointed; of course, if the kit or code change (note, not the supplier), Red Hat's prior certification may not now be accurate and so it is reasonable that Red Hat should not necessarily honour its past reliability statement.

Software maintenance ruling

As above, everyone has access to the same updates, upgrades, corrections and bug fixes as Red Hat is referring to. What nobody but Red Hat can access is any of its proprietary technology components that are usually located on the customer site. So Red Hat is well within the scope of any reasonable licence to prohibit another from utilising Red Hat's proprietary software. But this begs the question as to why any capable third-party supplier would want to use Red Hat's own tools when it can provide either its own, or utilise suitable open source solutions that achieve the same function.

Community-based healing process

While important enough to be considered in its own right, security remains part and parcel of software maintenance. Of course, given that the software being referred to here is Linux, Red Hat is by no means the only company or entity that provides rapid response solutions. Indeed, it is the very nature of open source software that code fixes to an open source product, like Linux, are available to the community. Where an entity other than Red Hat creates the security solution, Red Hat's customers would expect to benefit from this fix. This community-based healing process should, ideally, make fixes available to all customers, at exactly the same time, whether they be Red Hat's, Scientific Linux's, Oracle's or even a self-supported customer. There should be no exclusivity in code changes.

Production support

Those customers running mission-critical systems need 24/7 access to technical support engineers, either located on site or working remotely in telephone support

> "It is the very nature of FOSS software that code fixes are available to the community"

The old bait 'n' switch

Each point below deserves some consideration from a legal and commercial perspective as we look at the impact of switching service suppliers

Question: If I don't renew my subscription, can I continue to use the software?

Yes, under the General Public License (GPL), you may continue to use Red Hat software. However, for as long as you have any active subscriptions within your organisation, you are also bound by the terms of the Red Hat Enterprise Agreement. This agreement requires that as long as you have an active subscription, you are required to purchase a subscription for each system on which you are running Red Hat Enterprise Linux or JBoss Enterprise Middleware.

If you have no active subscriptions, you may still continue to use the software, but you will no longer receive…

Software access: access to the latest versions of Red Hat software that have been certified by thousands of independent software and hardware vendors.

Software maintenance: access to updates, upgrades, corrections and bug fixes for the software.

Security: rapid response to potential software security issues.

Production support: access to Red Hat technical support.

Open source assurance: participation in Red Hat's intellectual property assurance program.

centres. Support engineers normally own customer incidents through to resolution, often working with colleagues and the community to reproduce and solve critical technical issues. It is clear from the large cost differentials between the different levels of Red Hat support subscriptions offerings (eg Basic against Premium) that these are the high-revenue streams that Red Hat will be seeking to protect.

This explains why it discourages customers from looking at alternatives. That said, the fact will stand that either other support providers can offer similar capabilities and services or they cannot. Ultimately a customer will be able to evaluate whether the alternative supplier is stronger or weaker than Red Hat and then be able to take that into account on pricing. Customers are, however, not contractually bound to stay with any distro merely because it was the first installer.

Open source assurance

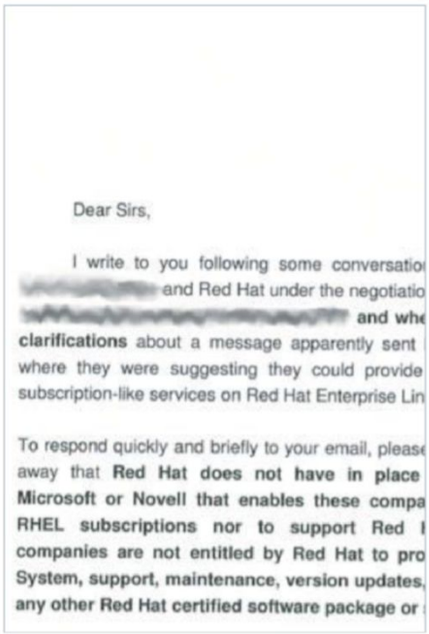
This is one aspect of Red Hat's withdrawals that does have a potential legal impact on the customer. CIOs will know that Linux, like all computer code, can infringe a third party's software patents and other intellectual property protections. Most readers will recall that in 2004, the Open Source Risk Management Group reported that there is a patent infringement risk that Linux users and developers should be aware of.

Merely because the software is free, that does not mean it has not copied another's code or does not infringe another's protected invention. As a user, therefore, it is a little troubling to know that installing Linux may be free of charge but not free of liability. In contrast, most proprietary software companies provide an indemnity to their users such that the supplier stands in the user's shoes if litigation ensues. Red Hat's open source assurance, to an extent, provides that in relation to the Linux code it supplies. In other words, the user should not be overly concerned that they will be out of pocket should an intellectual property action be brought against them. Customers would therefore be well advised to ensure that any substitute supplier provides them with the same or greater comfort.

Approach to moving customers

Above is an extract of a letter written by Red Hat to a customer seemingly involved in negotiations with Red Hat to change support provider. This extract was published by CNET in a blog article by Matt Asay, chief operating officer at Canonical.

> "Customers are not contractually bound to stay with any distro merely because it was the first installer"



Dear Sirs,

I write to you following some conversatio
and Red Hat under the negotiatio
and whe
clarifications about a message apparently sent
where they were suggesting they could provide
subscription-like services on Red Hat Enterprise Lin

To respond quickly and briefly to your email, pleas
away that Red Hat does not have in place
Microsoft or Novell that enables these compa
RHEL subscriptions nor to support Red
companies are not entitled by Red Hat to pro
System, support, maintenance, version updates,
any other Red Hat certified software package or

An extract of Red Hat's letter to a customer negotiating to change support provider

The clear message being delivered to the customer, emboldened by Red Hat, is that no other supplier can support Red Hat Enterprise Linux. In June, Red Hat pushed this concept even further in a newsletter to its 'partners' saying: "If [customers] choose not to renew, they must de-install Red Hat Enterprise Linux software from servers with expired subscriptions."

Note: These messages are neither technically true nor legally sustainable.

Technically, every byte in Red Hat Enterprise Linux is known by and usable by the community, which includes Red Hat, the customer and any supplier. Some suppliers may be more expert than Red Hat in supporting certain aspects of the kernel, others may be significantly worse than Red Hat. But everyone can support that code base.

As to the law, simply tossing a logo of any kind onto some open source software does not make it proprietary to Red Hat. That is the bargain all suppliers make when they decide to distribute Linux software: they do not own it. It follows that as they do not own it, they also have no rights to stop others supporting it, maintaining it, or providing bug fixes or patches to it.

Yes, they have rights to any proprietary elements, but these are likely to be removed by an incoming supplier in any event. What makes Red Hat's adoption of the tactics described in this article so interesting is that the vast proportion of the millions of lines of code that Red Hat distributes is not exclusive to it. The very nature of the Linux kernel and open source ecosystem means that anyone can legally support Red Hat's version of Linux.

Of course, if Red Hat did own the copyright in the Linux kernel it would be perfectly entitled to bundle any services or other software with its product. The law does not stop non-dominant suppliers from bundling one product with another.

Some customers, naturally, will be concerned by letters such as Red Hat's above and will assume they have no choice except to renew subscriptions and agreements and stay with Red Hat. Those customers need not be so concerned since they are in a superbly strong position. Once they have met their payment obligations to Red Hat, they take the free lunch and never need to pay Red Hat again.

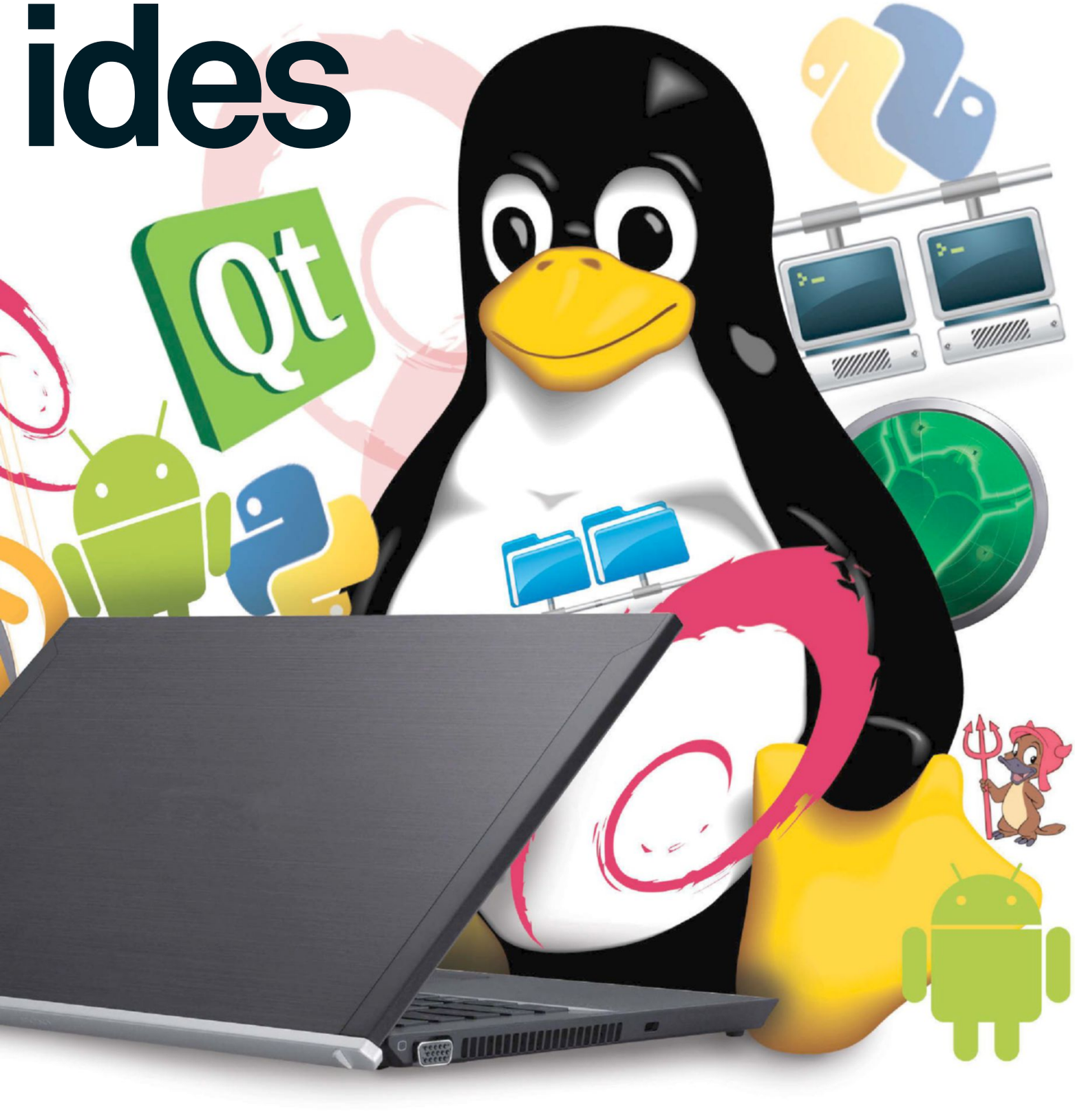
Written by Clive Gringras from Olswang's technology group with additional notes by Adrian Bridgwater

Developer gu

For system admins and coders

"Emacs has been called the best operating system masquerading as an editor"

ides



Get into system administration

In this article, we take a look at system administration in Linux and the various skills you'll need, including checking processing

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in IT for several years, concentrating on areas such as operating systems, networking, network security, eCommerce, internet services, LDAP and web servers

There are plenty of day-to-day administrative and maintenance procedures that system administrators need to follow to keep a Linux-based server or desktop system up and running. We can think of system administration as a bunch of things that we need to do to keep a computer system in usable order. Normally it includes things like installing new programs, creating accounts for users and deleting them when no longer needed, making sure that the files system is not corrupted, taking backup files and restoring them if necessary, and many other things. Another important system administration focus area is to monitor the system resources, the file system usage and the users. Some of the other things that a system administrator needs to be familiar with are starting and stopping the system, the booting process, run levels, init and its configuration etc. After reading this article, you should have a better understanding of what the role requires and the kinds of skills you will need to have.

> "If planning to administer the system, log in as root to perform the tasks"

1. Process management

In addition to the 'top' command, Linux commands such as 'pstree' and 'ps -auxw' will give detailed information about the various processes running on a system (**Fig 1**). As we know, every process is associated with a process ID, and the latter can be used as an argument to a 'kill' command to kill a specific process.

2. Monitor your system

It is very important to check the health of your Linux machine. Various commands will help us in system monitoring (**Fig 2**). Some of these commands are 'w' (it shows who is logged on and what they are doing), 'uptime' (it indicates how long the system has been running), and 'iostat' (it displays CPU statistics).

3. Monitoring the kernel modules

As a system administrator, one needs to focus on monitoring the loaded kernel modules too. Some of the basic commands that will serve the purpose here are 'cat /proc/modules' (it lists all currently loaded kernel modules) and '/sbin/lsmod'.

```
[swayam@centos1 ~]$ /sbin/lsmod
Module                  Size  Used by
vboxvfs                37064  0
autofs4                29253  3
hidp                   23105  2
rfcomm                 42457  0
l2cap                  29505  8 hidp,rfcomm
bluetooth              53925  5 hidp,rfcomm,
lockd                  63081  0
sunrpc                145405  2 lockd
ipv6                  267361  24
xfrm_nalgo             13381  1 ipv6
crypto_api             12609  1 xfrm_nalgo
dm_mirror              24393  0
dm_multipath           24909  0
scsi_dh                11713  1 dm_multipath
video                  21193  0
hwmon                   7365  0
backlight              10049  1 video
sbs                    18533  0
i2c_ec                  9025  1 sbs
button                 10705  0
battery                13637  0
asus_acpi              19289  0
ac                      9157  0
lp                     15849  0
floppy                 57125  0
snd_intel8x0           35421  0
snd_ac97_codec         93025  1 snd_intel8x0
ac97_bus                6337  1 snd_ac97_cod
snd_seq_dummy           7877  0
snd_seq_oss            32577  0
snd_seq_midi_event     11073  1 snd_seq_oss
```

■ Usage of the lsmod command

4. Changing the hostname

Sometimes it's necessary to change the hostname of a machine. The command 'hostname' displays a system's node name (**Fig 3**). Take a look at the contents of two important files on Linux: /etc/sysconfig/network and /etc/hosts.

```
[swayam@centos1 ~]$ ps -auxw
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2   2068   608 ?        Ss    2010   8:38 init [5]

root         2  0.0  0.0      0     0 ?        S<    2010   0:00 [migration/0]
root         3  0.0  0.0      0     0 ?        SN    2010   0:15 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S<    2010   4:49 [watchdog/0]
root         5  0.0  0.0      0     0 ?        S<    2010   5:29 [events/0]
root         6  0.0  0.0      0     0 ?        S<    2010   0:01 [khelper]
root         7  0.0  0.0      0     0 ?        S<    2010   0:00 [kthread]
root        10  0.0  0.0      0     0 ?        S<    2010   7:19 [kblockd/0]
root        11  0.0  0.0      0     0 ?        S<    2010   0:00 [kacpid]
root        47  0.0  0.0      0     0 ?        S<    2010   0:00 [cqueue/0]
root        50  0.0  0.0      0     0 ?        S<    2010   0:00 [khubd]
root        52  0.0  0.0      0     0 ?        S<    2010   0:00 [kseriod]
root       112  0.0  0.0      0     0 ?        S     2010   0:00 [pdflush]
root       113  0.0  0.0      0     0 ?        S     2010  16:57 [pdflush]
root       114  0.0  0.0      0     0 ?        S<    2010   9:15 [kswapd0]
root       115  0.0  0.0      0     0 ?        S<    2010   0:00 [aio/0]
root       274  0.0  0.0      0     0 ?        S<    2010   0:00 [kpsmoused]
root       306  0.0  0.0      0     0 ?        S<    2010   0:00 [ata/0]
root       307  0.0  0.0      0     0 ?        S<    2010   0:00 [ata_aux]
root       314  0.0  0.0      0     0 ?        S<    2010   0:00 [kstriped]
root       323  0.0  0.0      0     0 ?        S<    2010  10:45 [kjournald]
root       348  0.0  0.0      0     0 ?        S<    2010   0:40 [kauditd]
root       381  0.0  0.2   2284   532 ?        S<s   2010   0:00 /sbin/udevd -d
apache     484  0.0  3.7  50664  9560 ?        S     Jan02  0:00 /usr/sbin/httpd
apache     485  0.0  3.7  50664  9560 ?        S     Jan02  0:00 /usr/sbin/httpd
apache     486  0.0  3.7  50664  9560 ?        S     Jan02  0:00 /usr/sbin/httpd
apache     487  0.0  3.7  50664  9560 ?        S     Jan02  0:00 /usr/sbin/httpd
apache     488  0.0  3.7  50664  9560 ?        S     Jan02  0:00 /usr/sbin/httpd
apache     489  0.0  3.7  50664  9560 ?        S     Jan02  0:00 /usr/sbin/httpd
```

Fig 1 Process management A detailed look at process parameters

```
[swayam@centos1 ~]$
[swayam@centos1 ~]$ w
 07:40:26 up 34 days, 18:27,  2 users,  load average: 0.18, 0.07, 0.01
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
swayam   pts/2    129.221.5.202     06:58   41:49   0.10s  0.10s -bash
swayam   pts/3    129.221.5.202     07:20    0.00s  0.34s  0.00s w
[swayam@centos1 ~]$ uptime
 07:40:28 up 34 days, 18:27,  2 users,  load average: 0.18, 0.07, 0.01
[swayam@centos1 ~]$ iostat
Linux 2.6.18-164.el5 (centos1.centos.com)       01/03/2011

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.51    0.05    3.55    0.28    0.00   95.60

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               3.01         5.52        55.23   16634264  166560138
hda1              0.00         0.00         0.00       2954        110
hda2              0.33         0.62         8.11    1882730   24445080
hda3              2.11         2.95        39.04    8893731  117724146
hda4              0.00         0.00         0.00         10          0
hda5              0.57         1.92         8.07    5792373   24337610
hda6              0.00         0.01         0.00      15210       1456
hda7              0.00         0.00         0.01       6490      18312
hda8              0.00         0.01         0.01      40286      33424
hdc               0.00         0.00         0.00        152          0

[swayam@centos1 ~]$
```

Fig 2 Monitor your system Various commands on Linux to monitor a system

Fig 3 Changing the hostname Details displayed by hostname

```
[swayam@centos1 ~]$ hostname
centos1.centos.com
[swayam@centos1 ~]$
[swayam@centos1 ~]$ cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
#HOSTNAME=localhost.localdomain
HOSTNAME=centos1.centos.com
[swayam@centos1 ~]$
[swayam@centos1 ~]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
129.221.8.222   centos1.centos.com centos1
[swayam@centos1 ~]$
```

5. Information on file access and ownership

Linux provides various commands that are useful in getting information about files. The 'chmod' command is used to change the access permissions of a file. If we need to change the owner and group of a file, then we can use 'chown'. Another command, 'fuser', is used to identify processes using files or sockets (**Fig 4**).

6. Processes attached to open files

If we need to display the list of processes attached to open files, then we can use the 'lsof' command. By specifying the filename as a parameter, lsof displays the list of processes attached to a given file. 'lsof -u uid' lists all files open by a specific user UID.

7. Restricting user resources

Sometimes it is essential to control shell and process resources. The 'ulimit' command is used for this purpose. Using 'ulimit -a', one can display the limits of a shell. 'slt' is a good idea to see the limits assigned in /etc/security.

```
[swayam@centos1 ~]$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 4095
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 10240
cpu time               (seconds, -t) unlimited
max user processes              (-u) 4095
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
[swayam@centos1 ~]$
```

■ Various limits of a shell

```
cvs@suse-cvs:~> fuser
No process specification given
Usage: fuser [ -a | -s | -c ] [ -n SPACE ] [ -SIGNAL ] [ -kimuv ] NAME...
             [ - ] [ -n SPACE ] [ -SIGNAL ] [ -kimuv ] NAME...
       fuser -l
       fuser -V
Show which processes use the named files, sockets, or filesystems.

    -a        display unused files too
    -c        mounted FS
    -f        silently ignored (for POSIX compatibility)
    -i        ask before killing (ignored without -k)
    -k        kill processes accessing the named file
    -l        list available signal names
    -m        show all processes using the named filesystems
    -n SPACE  search in this name space (file, udp, or tcp)
    -s        silent operation
    -SIGNAL   send this signal instead of SIGKILL
    -u        display user IDs
    -v        verbose output
    -V        display version information
    -4        search IPv4 sockets only
    -6        search IPv6 sockets only
    -         reset options
```

Fig 4 Information on file access and ownership A close look at fuser

8. Process scheduling priority

One can modify the process scheduling priority using the 'nice' command (**Fig 5**). The range of priorities varies from -20 to 19 (-20 being the highest priority and 19 being the lowest). The default value for 'nice -n' is 10.

9. Monitor memory usage

Various commands on UNIX/Linux will help us to monitor memory usage (**Fig 6**). Some popular commands are 'vmstat' (to monitor virtual memory), 'free' (to display the amount of free and used memory in the system), 'pmap' (to examine the memory map and libraries) and 'sar'.

10. Monitor the file system

Monitoring the file system is another important task of system administrators. Linux provides a bunch of commands that can be used for monitoring and reporting the file system usage (**Fig 7**). Some of them are 'df -k' (to report file system disk space usage), 'du -sh' (to calculate file space usage for a given directory) and 'mount' (to display all mounted devices). The readers are advised to take a detailed look at the man pages of 'df' and 'mount'.

```
[swayam@centos1 ~]$ nice --help
Usage: nice [OPTION] [COMMAND [ARG]...]
Run COMMAND with an adjusted niceness, which affects process scheduling.
With no COMMAND, print the current niceness.  Nicenesses range from
-20 (most favorable scheduling) to 19 (least favorable).

  -n, --adjustment=N   add integer N to the niceness (default 10)
      --help     display this help and exit
      --version  output version information and exit

NOTE: your shell may have its own version of nice, which usually supersedes
the version described here.  Please refer to your shell's documentation
for details about the options it supports.

Report bugs to <bug-coreutils@gnu.org>.
[swayam@centos1 ~]$
```

Fig 5 Process scheduling priority Learn more about the 'nice' command

11. Synchronising files

Sometimes system administrators need to synchronise files on two different computers. This is pretty much required for website maintenance. The command 'rsync' can be used to synchronise files on two separate computers.

12. A look at system log files

Linux provides various system logs and a careful look at these log files will help an administrator in debugging the issues. **Some of the important system log files are:**

```
/var/log/messages
/var/log/secure
/var/log/maillog
```

A command, 'lastlog', will print the timestamp of the last login of various system users.

13. Rotating the log files

Many system and application programs generate log files. If left unchecked they would grow large enough to burden the system and application. Linux provides a program known as 'logrotate' and it will periodically back up the log file by

```
[swayam@centos1 ~]$ vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu---
-
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 0  0   8228  26068   5476  96220    0    0     3    28    9   10  1  4 96  0
[swayam@centos1 ~]$
[swayam@centos1 ~]$ free
             total       used       free     shared    buffers     cached
Mem:        255452     229444      26008          0       5476      96220
-/+ buffers/cache:     127748     127704
Swap:       522072       8228     513844
[swayam@centos1 ~]$
[swayam@centos1 ~]$ pmap
Usage: pmap [-x | -d] [-q] pid...
-x  show details
-d  show offset and device number
-q  quiet; less header/footer info
-V  show the version number
[swayam@centos1 ~]$
```

```
[swayam@centos1 ~]$ df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda2              7936288    668040   6858592   9% /
/dev/hda7               988088     17840    919244   2% /home
/dev/hda6              1984016     52224   1829380   3% /opt
/dev/hda5              2972236    256112   2562708  10% /var
/dev/hda3              4956316   3976604    723880  85% /usr
/dev/hda1               101086     11480     84387  12% /boot
tmpfs                   127724         0    127724   0% /dev/shm
[swayam@centos1 ~]$
[swayam@centos1 ~]$ du -sh
92K     .
[swayam@centos1 ~]$
[swayam@centos1 ~]$ mount
/dev/hda2 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda7 on /home type ext3 (rw)
/dev/hda6 on /opt type ext3 (rw)
/dev/hda5 on /var type ext3 (rw)
/dev/hda3 on /usr type ext3 (rw)
/dev/hda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
```

Fig 6 Monitor memory usage Various memory usage commands in the Linux environment

Fig 7 Monitor the file system Commands to monitor the file system

renaming it. The configuration file of importance here is /etc/logrotate.conf .

```
[swayam@centos1 ~]$ cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    minsize 1M
    create 0664 root utmp
    rotate 1
```

Take a look the configuration file for the logrotate tool

14. How to schedule a single occurrence of a task

The 'at' command can be used to schedule single jobs. The daemon 'atd' located at /usr/sbin will run the jobs scheduled with the 'at' command (**Fig 8**). The commands to be run are read from a file specified with the -f option, or from stdin if -f is not used.

15. Archiving utilities

Some of the useful packing and archiving utilities available on Linux are:

a. shar – It is used to create shell archives.

b. ar – This is another popular utility and is used to create, modify and extract from archive library files. Programmers use this utility to generate link libraries.

c. tar – This is the most widely used one. Developers normally use this for dual purposes – for archiving as well as for simple backups (**Fig 9**, overleaf). Please note that while archiving, tar preserves the file system information. One can also apply a compression utility to a tar file and it produces a compressed tar file (typically with an extension .tar.gz).

```
cvs@suse-cvs:~> at -f test.sh -v 9:28
Wed Jan  5 09:28:00 2011

warning: commands will be executed using /bin/sh
job 2 at 2011-01-05 09:28
Can't open /var/run/atd.pid to signal atd. No atd running?
cvs@suse-cvs:~>
```

Fig 8 How to schedule a single occurrence of a task Not a successful 'at' command (since atd was not running)

16. cron – the system administrator's favourite

There are situations where we need to schedule a recurring task. The 'cron' utility is used for this purpose. If we need to have a very specific schedule, then we need to add a line to the /etc/crontab file. You can also take a look at /var/spool/cron (if you have root privileges).

17. File compression and decompression utilities

Linux provides various file compression and decompression utilities. Important ones are 'zip' (for compressing), 'unzip' (for decompressing), 'gzip', 'gunzip', 'bzip2' and 'bunzip2' (**Fig 10**, overleaf). Their usage is pretty straightforward. All these utilities have well-defined man pages.

18. User information

On UNIX/Linux, the basic user database is a text file called /etc/passwd. This file has all valid usernames and their associated information. This file has one line per username. Most Linux systems have shadow passwords; in this case, the encrypted password is stored in a separate file /etc/shadow, which only root can read.

19. Monitoring user information

Linux provides various commands that can be used to monitor the user information. Some of them are 'who', 'w', 'users', 'groups', 'id' and 'last'. Another useful command is 'set' and it displays all the environmental variables in your current environment.

20. Viewing network connections

Linux provides an interesting command, 'netstat', that can be used to show network connections (**Fig 11**, overleaf). With the help of this, you'll be able to check the ports your system is listening on and the programs behind those ports.

```
cvs@suse-cvs:~> pwd
/home/cvs
cvs@suse-cvs:~> ls
bin        Documents     sample.txt  test_file_out    test.sh
Desktop    public_html   test_file   test_input_file
cvs@suse-cvs:~>
cvs@suse-cvs:~> tar -cvf test.tar bin
bin/
cvs@suse-cvs:~> ls -l test.tar
-rw-r--r-- 1 cvs users 10240 2011-01-04 15:46 test.tar
cvs@suse-cvs:~>
```

Fig 9 Archiving utilities How to create a tar ball

```
cvs@suse-cvs:~> ls -l test_input_file
-rw-r--r-- 1 cvs users 66 2010-09-22 13:48 test_input_file
cvs@suse-cvs:~> gzip test_input_file
cvs@suse-cvs:~>
cvs@suse-cvs:~> ls -l *.gz
-rw-r--r-- 1 cvs users 96 2010-09-22 13:48 test_input_file.gz
cvs@suse-cvs:~> gunzip test_input_file.gz
cvs@suse-cvs:~>
cvs@suse-cvs:~> ls -l test_input_file
-rw-r--r-- 1 cvs users 66 2010-09-22 13:48 test_input_file
cvs@suse-cvs:~>
```

Fig 10 File compression and decompression utilities Usage of gzip and gunzip utilities

21. Update local host aliases

For updating local host aliases, we need to edit the file /etc/hosts. Please note that you need to be a root user in order to edit this file. **The format to be used is given below:**

```
IP_Address <tab> www.your.domain
<tab> alias
```

FOR LOCALHOST, THIS WILL LOOK LIKE:

```
127.0.0.1 <tab> localhost.
localdomain <tab> localhost
```

22. Displaying system information

One can use the 'uname -a' command to display the system information. Linux also provides another useful command, 'dmesg', which can be used to display bootup messages. Take a look at its man page to learn more on it.

```
cvs@suse-cvs:~> dmesg | more
Linux version 2.6.16.46-0.12-default (geeko@buildhost) (gcc
15 (prerelease) (SUSE Linux)) #1 Thu May 17 14:00:09 UTC 200
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)
 BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000001fef0000 (usable)
 BIOS-e820: 000000001fef0000 - 000000001feff000 (ACPI data)
 BIOS-e820: 000000001feff000 - 000000001ff00000 (ACPI NVS)
 BIOS-e820: 000000001ff00000 - 0000000020000000 (usable)
 BIOS-e820: 00000000fec00000 - 00000000fec10000 (reserved)
 BIOS-e820: 00000000fee00000 - 00000000fee01000 (reserved)
 BIOS-e820: 00000000fff00000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
512MB LOWMEM available.
found SMP MP-table at 000f6cd0
On node 0 totalpages: 131072
  DMA zone: 4096 pages, LIFO batch:0
  DMA32 zone: 0 pages, LIFO batch:0
  Normal zone: 126976 pages, LIFO batch:31
  HighMem zone: 0 pages, LIFO batch:0
MI present.
```

■ Bootup message of a system

23. IPC clean-up

Many processes on your system may use various inter-process communication (IPC) and thus there is always a need to clean them up manually. One can use the 'ipcs' command to view various IPCs and the 'ipcrm' command to remove the IPC. Take a look at their man pages to understand more.

```
cvs@suse-cvs:~> netstat -ape | more
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User       Inode      PID/Program name
tcp        0      0 *:cvspserver            *:*                     LISTEN
root       8368       -
tcp        0      0 *:57218                 *:*                     LISTEN
root       8373       -
tcp        0      0 *:printer               *:*                     LISTEN
root       8367       -
tcp        0      0 *:wbem-https            *:*                     LISTEN
root       9460       -
```

Fig 11 Viewing network connections Use netstat command to view network connections

24. Know how to shut down a machine

For proper shutting down of a Linux machine, a command – 'shutdown' – can be used. If your system has many users, we need to use the command 'shutdown -h +time message', where 'time' is the time in minutes until the system is halted, and 'message' is a short explanation of why the system is shutting down.

init and its configuration

'init' is the first user-level process started by the kernel on UNIX/Linux. **Some of the important functionalities of the init process are:**

a. Starting getty so that users will be able to log in.

b. Taking care of orphaned processes.

Typically, the kernel looks for init in a few locations, but the correct location is /sbin/init. When init gets started, it completes the boot process by carrying out several administrative tasks. Some of these tasks include the checking of the file system, starting of various services, starting a getty for each terminal etc. As we have already mentioned above, init adopts orphan processes. When a process starts a child process and dies before its child, the child becomes parentless. Such a child immediately becomes a child of init.

Let us see how we can configure init to start getty. When it starts up, init reads a configuration file, /etc/inittab. Let us take a look at the contents of this configuration file.

As can be seen on screen (**Fig 12**), each entry in that configuration file consists of four fields – ID, run levels, action and process. Please note that the ID basically specifies the line in the file. For getty lines, it specifies the terminal it runs on

Let us say we want to run getty on the first virtual terminal (ie /dev/tty1). **Then we will have an entry as shown below:**

```
1:2345:respawn:/sbin/getty 9600 tty1
```

In the above example, the first field says that this is the line for /dev/tty1. The second field says that it applies to run levels 2, 3, 4, and 5. The third field indicates that the command should be run again after it exits (this will basically allow one to log in, log out and then log in again). The last field is the command that runs getty on the first virtual terminal.

A run level is a software configuration of the

```
[root@centos1 etc]# cat inittab
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Fig 12 init and its configuration A look at the contents of the /etc/inittab file

system that allows only a selected group of processes to exist. **The following table takes a look at various run levels:**

0	Halt a system
1	Single user mode
2	Multi-user without network services
3	Full multi-user with networking
4	Not used
5	Full multi-user with networking and X Window (GUI)
6	Reboot

A run level of 1 is referred to as single-user mode. This single-user mode is necessary for some administrative tasks.

While configuring /etc/inittab, one can use some special keywords (in the third field). When we have such keywords, init will be able to respond to special circumstances. **The popular special keywords are:**

a. powerwait – When power fails, this allows init to shut down the system.

b. ctrlaltdel – This allows init to reboot the system whenever Ctrl+Alt+Del is pressed.

c. sysinit – This is the command to be run when the system is booted.

The above list is not a complete one. Please refer to the man page of inittab to learn about other special keywords.

Therefore we now understand that the kernel will start init. This process reads the file /etc/inittab and uses this file to determine how to create processes. It is important to note here that init is always running and can dynamically do things.

Backups made simple

We all know that it is critical to back up our data available on a machine. We may lose data because of several reasons – some of these reasons are hardware failures, software bugs and natural disasters. It is essential that we carry out this backup procedure properly and regularly. A lot of attention needs to be given when we are selecting the backup medium. Some of the parameters that need to be considered here are availability, reliability, cost and speed.

Before we back up the data, we need to have a proper plan. This is the phase where we will determine which files are unnecessary to back up, which files are irreplaceable without a backup, who owns the backup process etc. It is natural to expect the root user to perform the backup procedure since only the root user has full permissions.

There are various types of backups. **Some of them are listed here:**

a. Full backup – Here we back up all files and folders. As expected, this requires the highest storage space requirement.

b. Incremental backup – We back up everything that has changed since our last backup.

c. Network backup – It means backing up a client to a backup server. In this, the client sends the files to a server and the server writes them to a backup medium.

Now let's take a look at some of the tools that one can use for taking simple backups...

a. tar utility – As we have mentioned above, one can use tar as a standard backup tool. It always reads the backup volume sequentially, so for large volumes it is very slow.

b. cpio – This is considered as one of the most flexible commands. You need to understand its man page completely before you start using it. Please take a look at the way in which the command must be entered.

c. dd – This command is one of the original UNIX utilities and it finds extensive use in backup functionalities.

d. dump – This utility is specifically written for backups. It reads the file system directly.

Both tar and cpio are capable of storing files on tapes, and retrieving files from them. And both of them are capable of using almost any media.

Let us understand how we can back up the current directory to a tape using tar:

```
$tar -cv .                    (Assuming
that the default tape drive is rsa0)
$tar -cvf /dev/rsa1 .    (And second
tape drive as rsa1)
```

We can extract (or restore) files from tape using the following commands:

```
$tar -xv .
$tar -xvf /dev/rsa1
```

With the cpio command, for backup, we need to use it with a search command (typically 'find').

In its simplest form, the following command can be used to read filenames using the 'find' command and then copy them to the floppy drive /dev/fd0

```
$ find . -print | cpio -ocv > /dev/fd0
```

We can selectively restore the files using the '-i' option, as indicated below

```
$ cpio -i "*.f" "*.c" </dev/fd0
```

In this article, we have taken a look at the various aspects of system administration. Now it is time for you to go and use them in your day-to-day operations.

Python for system administrators

Learn how Python can help in system administration as it dares to replace the usual shell scripting...

Resources

Search your distribution's package repository to install the following packages...

Python-devel Python development libraries, required for compiling third-party Python module

setuptools setuptools allows you to download, build, install, upgrade, and uninstall Python packages with ease

System administration is an important part of our computing environment. It does not matter whether you are managing systems at your work our home. Linux, being a UNIX-based operating system, already has everything a system administrator needs, such as the world-class shells (not just one but many, including Bash, csh, zsh etc), handy tools, and many other features which make the Linux system an administrator's dream. So why do we need Python when Linux already has everything built-in? Being a dynamic scripting language, Python is very easy to read and learn. That's just not us saying that, but many Linux distributions actually use Python in core administrative parts. For example, Red Hat (and Fedora) system setup tool Anaconda is written in Python (read this line again, got the snake connection?). Also, tools like GNU Mailman, CompizConfig Settings Manager (CCSM) and hundreds of tiny GUI and non-GUI configuration tools are written using Python. Python does not limit you on the choice of use interface to follow – you can build command-line, GUI and web apps using Python. This way, it has got covered almost all the possible interfaces.

In this feature we will look into executing sysadmin-related tasks using Python.

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Note

This feature assumes that you are familiar with basic Python. We will cover a few basics wherever required. The Python development masterclass published in issue 88 of **Linux User & Developer** would be a good reference to gain familiarity with Python. It is also available online at **www.linuxuser.co.uk/tutorials/python-development-masterclass/**.

Parsing configuration files

Configuration files provide a way for applications to store various settings. In order to write a script that allows you to modify settings of a particular application, you should be able to parse the configuration file of the application. In this section we learn how to parse INI-style configuration files. Although old, the INI file format is very popular with much modern open source software, such as PHP and MySQL.

Excerpt for php.ini configuration file:

```
[PHP]
engine = On
zend.ze1_compatibility_mode = Off
short_open_tag = On
asp_tags = Off
precision    =  14
y2k_compliance = On
output_buffering = 4096
;output_handler =
zlib.output_compression = Off

[MySQL]
; Allow or prevent persistent links.
mysql.allow_persistent = On
mysql.max_persistent = 20
mysql.max_links = -1
mysql.default_port = 3306
mysql.default_socket =
mysql.default_host = localhost
mysql.connect_timeout = 60
mysql.trace_mode = Off
```

Python provides a built-in module called ConfigParser (known as configparser in Python 3.0). You can use this module to parse and create configuration files.

@code: writeconfig.py
@description: The following demonstrates adding MySQL section to the php.ini file.
@warning: Do not use this script with the actual php.ini file, as it's not designed to handle all aspects of a complete php.ini file.

```
import ConfigParser
config = ConfigParser
RawConfigParser()

config.add_section('MySQL')
config.set('MySQL','mysql.trace_
mode','Off')
config.set('MySQL','mysql.connect_
timeout','60')
config.set('MySQL','mysql.default_
host','localhost')
config.set('MySQL','mysql.default_
port','3306')
config.set('MySQL','mysql.allow_
persistent', 'On' )
config.set('MySQL','mysql.max_
persistent','20')

with open('php.ini', 'ap') as
configfile:
    config.write(configfile)

Output:php.ini
[MySQL]
mysql.max_persistent = 20
mysql.allow_persistent = On
mysql.default_port = 3306
mysql.default_host = localhost
mysql.trace_mode = Off
mysql.connect_timeout = 60
```

Note

This feature is written for the Python 2.X series, as it is still the most popular and default Python distribution across all the platforms (including all Linux distros, BSDs and Mac OS X).

@code: parseconfig.py
@description: Parsing and updating the config file

```
import ConfigParser
config = ConfigParser.ConfigParser()
config.read('php.ini')
# Print config values
print config.get('MySQL','mysql.
default_host')
print config.get('MySQL','mysql.
default_port')
config.remove_option('MySQL','mysql.
trace_mode')
with open('php.ini', 'wb') as
configfile:
    config.write(configfile)
```

Parsing JSON data

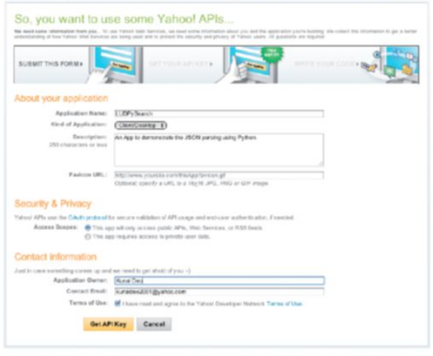
JSON (also known as JavaScript Object Notation) is a lightweight modern data-interchange format. JSON is an open standard under ECMA-262. It is a text format and is completely language-independent. JSON is also used as the configuration file format for modern applications such as Mozilla Firefox and Google Chrome. JSON is also very popular with modern web services such as Facebook, Twitter, Amazon EC2 etc. In this section we will use the Python module 'simplejson' to access Yahoo Search (using the Yahoo Web Services API), which outputs JSON data.

To use this section, you should have the following:

1. Python module: simplejson.

Note: You can install Python modules using the command 'easy_install <module name>'. This command assumes that you have a working internet connection.

2. Yahoo App ID: The Yahoo App ID can be created from **https://developer.apps.yahoo.com/dashboard/createKey.html**. The Yahoo App ID will be generated on the next page. See the screenshot below for details.



Generating the Yahoo App ID

simplejson is very easy to use. In the following example we will use the capability of mapping JSON data structures directly to Python data types. This gives us direct access to the JSON data without developing any XML parsing code.

JSON PYTHON DATA MAPPING

JSON	Python
object	dict
array	list
string	unicode
number (int)	int, long
number (real)	float
TRUE	TRUE
FALSE	FALSE
null	None

For this section we will be using the simplejson.load function, which allows us to deserialise a JSON object into a Python object.

@code: LUDSearch.py

```
import simplejson, urllib
APP_ID = 'xxxxxxxx' # Change this to
your APP ID
SEARCH_BASE = 'http://search.
yahooapis.com/WebSearchService/V1/
webSearch'

class YahooSearchError(Exception):
    pass

def search(query, results=20,
start=1, **kwargs):
    kwargs.update({
        'appid': APP_ID,
        'query': query,
        'results': results,
        'start': start,
        'output': 'json'
    })
    url = SEARCH_BASE + '?' +
urllib.urlencode(kwargs)
    result = simplejson.load(urllib.
urlopen(url))
    if 'Error' in result:
        # An error occurred; raise
an exception
        raise YahooSearchError,
result['Error']
    return result['ResultSet']
```

Now let's use the above code from the Python shell to see how it works. Change to the directory where you have saved the LUDYSearch.py and open a Python shell.

@code: Python Shell Output. Lines starting with '>>>' indicate input

```
>>> execfile("LUDYSearch.py")
>>> results = search('Linux User and
Developer')
>>> results['totalResultsAvailable']
123000000
>>> results['totalResultsReturned']
20
>>> items = results['Result']
>>> for Result in items:
...     print Result['Title'],Resul
t['Url']
...
Linux User http://www.linuxuser.
co.uk/
Linux User and Developer -
Wikipedia, the free encyclopedia
 http://en.wikipedia.org/wiki/Linux_
User_and_Developer
Linux User &amp;amp; Developer |
Linux User http://www.linuxuser
.co.uk/tag/linux-user-developer/
```

Gathering system information

One of the important jobs of a system administrator is gathering system information. In this section we will use the SIGAR (System Information Gatherer And Reporter) API to demonstrate how we can gather system information using Python. **SIGAR is a very complete API and it can provide lot of information, including the following:**

1. System memory, swap, CPU, load average, uptime, logins.
2. Per-process memory, CPU, credential info, state, arguments, environment, open files.
3. File system detection and metrics.
4. Network interface detection, configuration info and metrics.
5. TCP and UDP connection tables.
6. Network route table.

Installing SIGAR

The first step is to build and install SIGAR. SIGAR is hosted at GitHub, so make sure that you have Git installed in your system.

Then perform the following steps to install SIGAR and its Python bindings:

```
$ git clone git://github.com/
hyperic/sigar.git sigar.git
$ cd sigar.git/bindings/python
$ sudo python setup.py install
```

At the end you should see a output similar to the following:

```
Writing /usr/local/lib/python2.6/
dist-packages/pysigar-0.1.egg-info
```

SIGAR is a very easy-to-use library and can be used to get information on almost every aspect of a system. The next example shows you how.

The following code shows the memory and the file system information

@code: PySysInfo.py

```
import os
import sigar
sg = sigar.open()
mem = sg.mem()
swap = sg.swap()
fslist = sg.file_system_list()
print "==========Memory
Information=============="
print "\tTotal\tUsed\tFree"
print "Mem:\t",\
    (mem.total() / 1024), \
    (mem.used() / 1024), \
    (mem.free() / 1024)
print "Swap:\t", \
    (swap.total() / 1024), \
    (swap.used() / 1024), \
    (swap.free() / 1024)
print "RAM:\t", mem.ram(), "MB"
print "==========File System
Information=============="
def format_size(size):
    return sigar.format_size(size *
1024)
print 'Filesystem\tSize\tUsed\
tAvail\tUse%\tMounted on\tType\n'
for fs in fslist:
    dir_name = fs.dir_name(
    usage = sg.file_system_
usage(dir_name)
    total = usage.total()
    used = total - usage.free()
    avail = usage.avail()
    pct = usage.use_percent() * 100
    if pct == 0.0:
        pct = '-'
    print fs.dev_name(), format_
size(total), format_size(used),
format_size(avail),\
        pct, dir_name, fs.sys_type_
name(), '/', fs.type_name()
```

@Output

```
==========Memory
Information==============
   Total        Used    Free
Mem:    8388608 6061884 2326724
Swap:   131072 16048 115024
RAM:    8192 MB
==========File System
Information==============
Filesystem      Size    Used
Avail   Use%    Mounted on      Type
/dev/disk0s2 300G 175G 124G 59.0 /
hfs / local
devfs 191K 191K   0  - /dev devfs /
none
```

Accessing Secure Shell (SSH) services

SSH (Secure Shell) is a modern replacement for an old remote shell system called Telnet. It allows data to be exchanged using a secure channel between two networked devices. System administrators frequently use SSH to administrate networked systems. In addition to providing remote shell, SSH is also used for secure file transfer (using SSH File Transfer Protocol, or SFTP) and remote X server forwarding (allows you to use SSH clients as X server). In this section we will learn how to use the SSH protocol from Python using a Python module called paramiko, which implements the SSH2 protocol for Python.

paramiko can be installed using the following steps:

```
$ git clone https://github.com/
robey/paramiko.git
$ cd paramiko
$ sudo python setup.py install
```

To the core of paramiko is the SSHClient class. This class wraps L{Transport}, L{Channel}, and L{SFTPClient} to handle most of the aspects of SSH. **You can use SSHClient as**

```
client = SSHClient()
client.load_system_host_keys()
client.connect('some.host.com')
stdin, stdout, stderr = client.
exec_command('dir')
```

The following example demonstrates a full SSH client written using the paramiko module.

@code: PySSHClient.py

```
import base64, getpass, os, socket,
sys, socket, traceback
import paramiko
import interactive
# setup logging
paramiko.util.log_to_file('demo_
simple.log')
# get hostname
username = ''
if len(sys.argv) > 1:
    hostname = sys.argv[1]
    if hostname.find('@') >= 0:
        username, hostname = hostname.
split('@')
else:
    hostname = raw_input('Hostname: ')
if len(hostname) == 0:
    print '*** Hostname required.'
    sys.exit(1)
port = 22
if hostname.find(':') >= 0:
    hostname, portstr = hostname.
split(':')
    port = int(portstr)
# get username
if username == '':
    default_username = getpass.
getuser()
    username = raw_input('Username
[%s]: ' % default_username)
    if len(username) == 0:
        username = default_username
password = getpass.getpass('Password
for %s@%s: ' % (username, hostname))
# now, connect and use paramiko
Client to negotiate SSH2 across the
connection
try:
    client = paramiko.SSHClient()
    client.load_system_host_keys()
    client.set_missing_host_key_
policy(paramiko.WarningPolicy)
    print '*** Connecting...'
    client.connect(hostname, port,
username, password)
    chan = client.invoke_shell()
    print repr(client.get_transport())
    print '*** SSH Server Connected!
***'
    print
    interactive.interactive_
shell(chan)
    chan.close()
    client.close()
except Exception, e:
    print '*** Caught exception: %s:
%s' % (e.__class__, e)
    traceback.print_exc()
    try:
        client.close()
    except:
        pass
    sys.exit(1)
```

Note

If you are confused with the tab spacing of the code, refer to Python documentation.

To run this code you will also need a custom Python class interactive.py which implements the interactive shell for the SSH session. Look for this file online and copy it into the same folder where you have created PySSHClient.py .

@code_Output

```
kunal@ubuntu-vm-kdeo:~/src/paramiko/
demos$ python demo_simple.py
Hostname: 192.168.1.2
Username [kunal]: luduser
Password for luduser@192.168.1.2:
*** Connecting...
<paramiko.Transport at 0xb76201acL
(cipher aes128-ctr, 128 bits)
(active; 1 open channel(s))>
*** SSH Server Connected! ***
Last login: Thu Jan 13 02:01:06 2011
from 192.168.1.9
[~ $:]
```

If the host key for the SSH server is not added to your $HOME/.ssh/known_hosts file, the client will throw the following error:

```
*** Caught exception: <type
'exceptions.TypeError'>: unbound
method missing_host_key() must be
called with WarningPolicy instance
as first argument (got SSHClient
instance instead)
```

This means that the client cannot verify the authenticity of the server you are connected to. To add the host key to known_hosts, you can use the ssh command. It is important to remember that this is not the ideal way to add the host key; instead you should use ssh-keygen. But for simplicity's sake we are using the ssh client.

```
kunal@ubuntu-vm-kdeo:~/.ssh$ ssh
luduser@192.168.1.2
The authenticity of host
'192.168.1.2 (192.168.1.2)' can't be
established.
RSA key fingerprint is be:01:76:6a:
b9:bb:69:64:e3:dc:37:00:a4:36:33:d1.
Are you sure you want to continue
connecting (yes/no)? yes
Warning: Permanently added
'192.168.1.2' (RSA) to the list of
known hosts.
```

So now you've seen how easily we can carry out the complex sysadmin tasks using Python. As with all the Python code, all the presented code can easily be adopted into a GUI application (using PyGTK or PyQt) or a web app (using Django or Grok). In upcoming issues we will look into doing more with the Python scripting language.

Writing a user interface using Python

Learn how to create a user-friendly interface using Python

Administrators are comfortable with running raw scripts by hand, but end-users are not. So if you are writing a script that is supposed to be used by common users, it is a good idea to create a user-friendly interface on top of the script. This way end-users can run the scripts just like any other application. To demonstrate this, we will create a simple GRUB configuration tool which allows users to select default boot entry and the timeout. We will be creating a TUI (text user interface) application and will use the Python module 'snack' to facilitate this (not to be confused with the Python audio library, tksnack).

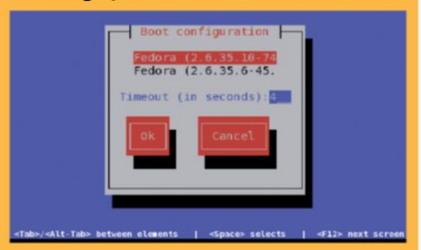
This app consists of two files…

grub.py: GRUB Config File (grub.conf) Parser. It implements two main functions, readBootDB() and writeBootFile(), which are responsible for reading and writing the GRUB configuration file.

grub_tui.py: Text user interface file for manipulating the GRUB configuration file using the functions available in grub.py.

@code:grub_tui.py

```
import sys
from snack import *

from grub import (readBootDB,
writeBootFile)

def main(entry_
value='1',kernels=[]):
    try:
        (default_value, entry_
value, kernels)=readBootDB()
    except:
        print >> sys.stderr,
("Error reading /boot/grub/grub.
conf.")
   sys.exit(10)

    screen=SnackScreen()

    while True:
        g=GridForm(screen, ("Boot
configuration"),1,5)
        if len(kernels)>0 :
            li=Listbox(heig
ht=len(kernels), width=20,
returnExit=1)
            for i, x in
enumerate(kernels):
                li.append(x,i)
            g.add(li, 0, 0)

li.setCurrent(default_value)

        bb = ButtonBar(screen,
((("Ok"), "ok"), (("Cancel"),
"cancel")))

        e=Entry(3, str(entry_
value))
        l=Label(("Timeout (in
seconds):"))
        gg=Grid(2,1)
        gg.setField(l,0,0)
        gg.setField(e,1,0)

        g.add(Label(''),0,1)
        g.add(gg,0,2)
        g.add(Label(''),0,3)
        g.add(bb,0,4,growx=1)
        result = g.runOnce()
        if
bb.buttonPressed(result) ==
'cancel':
            screen.finish()
            sys.exit(0)
        else:
            entry_value =
e.value()
            try :
                c = int(entry_
value)
                break
            except ValueError:
                continue

    writeBootFile(c,
li.current())
    screen.finish()

if __name__== '__main__':
    main()
```



Start the tool using the sudo command (as it reads the grub.conf file)

```
$ sudo grub_tui.py
```

Linux server monitoring made simple

In this article, Swayam Prakasha takes us through one of the most important system admin duties

Advisor

Swayam Prakasha has a master's degree in computer engineering. He has been working in iT for several years, concentrating on areas such as operating systems, networking, network security, eCommerce, and LDAP and web servers

Server monitoring is considered to be an important an activity as a system administrator needs to carry out. On Linux environment, various built-in commands and add-on tools are available for server monitoring. These commands and tools will make the life of system administrators much simpler. With the help of these tools, we will be able to gather information about system activities and we can also use them to find the possible causes whenever we experience a performance issue. They are helpful when we analyse and debug server issues such as bottlenecks in disk, memory, CPU, network etc. Linux-based servers are widely used by many enterprises today. Monitoring the server and critical parts of the operating system (such as memory, disk, network interfaces) is very much critical to make sure that the applications running on these servers are working efficiently at all times.

```
[root@centos1 ~]# top
top - 07:58:18 up 2 days, 17:05,  3 users,  load average: 0.00, 0.00, 0.00
Tasks: 128 total,   2 running, 126 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3%us,  0.3%sy,  0.0%ni, 99.0%id,  0.3%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    255452k total,   240276k used,    15176k free,    36800k buffers
Swap:   522072k total,      184k used,   521888k free,    84748k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      15   0  2068  624  532 S  0.0  0.2   0:07.35 init
    2 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.49 ksoftirqd/0
    4 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    5 root      10  -5     0    0    0 S  0.0  0.0   0:00.13 events/0
    6 root      10  -5     0    0    0 S  0.0  0.0   0:02.71 khelper
    7 root      17  -5     0    0    0 S  0.0  0.0   0:00.09 kthread
   10 root      15  -5     0    0    0 S  0.0  0.0   0:08.62 kblockd/0
   11 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
   47 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 cqueue/0
   50 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 khubd
   52 root      18  -5     0    0    0 S  0.0  0.0   0:00.00 kseriod
  112 root      15   0     0    0    0 S  0.0  0.0   0:00.72 pdflush
  113 root      15   0     0    0    0 S  0.0  0.0   0:59.09 pdflush
  114 root      10  -5     0    0    0 S  0.0  0.0   0:00.52 kswapd0
```

Fig 1 A look at the 'top' command in the Linux environment

In order to monitor various parameters of a Linux server, the operating system provides several useful built-in commands. System administrators are always interested in making the most of these commands and utilities so that server monitoring becomes easy to take care of. Work station monitoring may be a good starting point if one is interested in exploring server monitoring.

System administrators will always be interested in obtaining a real-time view of

> "In order to monitor various parameters of a Linux server, the operating system provides several useful built-in commands"

What parameters should I monitor?

When we talk about server monitoring, we need to focus our attention on a wide range of important parameters including:

a. CPU monitoring As far as CPU monitoring is concerned, we may be interested in knowing the top most CPU-consuming processes on a server. This will help an administrator in understanding which processes are causing a CPU spike on a server.

b. Memory monitoring The focus here is to get metrics such as available free memory, top processes that are consuming memory on a server, usage of swap memory etc. With these metrics, a system administrator can take necessary corrective actions as needed.

c. I/O monitoring Here the idea is to understand if there is any I/O bottleneck on a server. System administrators will always be interested in tracking the number of processes blocked on I/O.

d. Disk space monitoring Here, metrics of importance are related to disk space (such as total capacity and free space). It is always a good idea to track the free space on each of the disk partitions of a server.

e. Network traffic monitoring One needs to know the incoming and outgoing traffic so that network interfaces with maximum traffic can be identified.

f. Process monitoring Metrics of importance here are running processes, memory usage, CPU usage etc. With these metrics, one will be able to track the number of processes of a specific application that are running simultaneously, monitor the CPU usage of an application over time and track the memory usage of an application over time.

g. Server log monitoring With this, we will be able to get information about the events in various system log files. Some of the useful system log files are /var/adm/messages, syslog etc.

a running system. The real-time view will basically provide the actual process activity on a server. Linux operating system provides a command – top – that is widely used to gather information about the process activity. This command shows the most CPU-intensive tasks running on the server and updates the list every five seconds. Readers are advised to go through the detailed man page of this popular command.

The 'top' command is very useful in identifying CPU utilisation and it will help a system administrator to troubleshoot CPU-related performance issues. It can display system summary information as well as a list of tasks currently being managed by the Linux kernel. The 'top' command monitors CPU utilisation, process statistics and memory utilisation. As can be seen from the screenshot on the preceding page (**Fig 1**), the upper section contains information related to overall system status – uptime, load average, process counts, CPU status, and utilisation statistics for both memory and swap space. Note that by default, the processes are ordered by percentage of CPU usage. 'Top' is considered a very useful command for system administrators. This is basically because it shows which users and processes are consuming the most system resources at any given time.

If we are using an SMP (multiple CPU) system, we can use the mpstat command to display the utilisation of each CPU individually. This command basically reports processor-related statistics. System administrators use this command in monitoring activities so that they are able to diagnose problems. We take a look at its man page in **Fig 2**.

```
[root@centos1 ~]# sar
Linux 2.6.18-164.el5 (centos1.centos.com)       03/14/2011

12:00:01 AM       CPU     %user     %nice   %system   %iowait    %steal     %idl
e
12:30:01 AM       all      0.28      0.00      0.61      0.30      0.00     98.8
0
12:40:01 AM       all      0.27      0.00      0.61      0.27      0.00     98.8
5
01:20:01 AM       all      0.27      0.00      0.59      0.32      0.00     98.8
2
01:30:01 AM       all      0.27      0.00      0.60      0.31      0.00     98.8
1
02:00:01 AM       all      0.30      0.00      0.62      0.30      0.00     98.7
8
02:10:01 AM       all      0.25      0.00      0.56      0.28      0.00     98.9
1
02:20:01 AM       all      0.27      0.00      0.60      0.28      0.00     98.8
5
02:50:01 AM       all      0.22      0.00      0.53      0.29      0.00     98.9
6
03:00:01 AM       all      0.27      0.00      0.57      0.31      0.00     98.8
6
03:10:01 AM       all      0.28      0.00      0.60      0.32      0.00     98.8
```

Fig 4 Displaying sar output on our system

```
[root@centos1 ~]# man mpstat
Formatting page, please wait...
MPSTAT(1)                   Linux Userâs Manual                  MPSTAT(1)

NAME
       mpstat - Report processors related statistics.

SYNOPSIS
       mpstat [ -P { cpu | ALL } ] [ -V ] [ interval [ count ] ]

DESCRIPTION
       The mpstat command writes to standard output activities for each avail-
       able processor, processor 0 being the first one.  Global average activ-
       ities  among  all processors are also reported.  The mpstat command can
       be used both on SMP and UP machines, but in  the  latter,  only  global
       average activities will be printed.

       The  interval parameter specifies the amount of time in seconds between
       each report.  A value of 0 (or no parameters  at  all)  indicates  that
       processors  statistics  are  to  be  reported for the time since system
       startup (boot).  The count parameter can be  specified  in  conjunction
       with  the  interval parameter if this one is not set to zero. The value
       of count determines the number of reports generated at interval seconds
       apart. If the interval parameter is specified without the count parame-
```

Fig 2 A detailed look at the man page of mpstat

```
[root@centos1 ~]# mpstat
Linux 2.6.18-164.el5 (centos1.centos.com)       03/14/2011

08:34:03 AM  CPU   %user   %nice    %sys %iowait    %irq   %soft  %steal   %idle
    intr/s
08:34:03 AM  all    0.35    0.08    0.74    0.39    0.09    0.03    0.00   98.32
   1033.46
[root@centos1 ~]#
```

Fig 3 Execute mpstat on your Linux box

Please analyse the output by executing the mpstat command with the '-P ALL' option. Note that the 'ALL' keyword indicates that statistics are to be reported for all processors.

If we are interested in displaying today's CPU activity, then we can go for the sar command. The sar command is typically used to collect and report system activity information. Please refer to its man page to get complete details of the sar command. We have executed sar on our system and captured the output in the above screenshot (**Fig 4**).

A typical use of sar is to compare CPU utilisation. For example, 'sar -u 3 5' will display comparison of CPU utilisation – three seconds apart, five times. It is a good idea to take a look at the output of 'sar -u 3 5'. Sar helps in obtaining qualitative data about your system at periodic intervals and helps in finding performance bottlenecks and determining whether any specific action is needed.

Another important command that can be used to report CPU statistics and input / output statistics for devices, partitions and network file systems is iostat. An important metric provided by this command is the system's average CPU utilisation since the last reboot. We have captured the output of iostat execution in the above screenshot (**Fig 5**).

> **The pmap command is used to report a memory map of processes… it can be useful for finding out the causes of memory bottlenecks**

```
[root@centos1 ~]# iostat
Linux 2.6.18-164.el5 (centos1.centos.com)       03/14/2011

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.35    0.08    0.86    0.39    0.00   98.32

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               3.97        12.18        58.98    2808604   13593838
hda1              0.00         0.01         0.00       2494         18
hda2              0.63         1.62        11.57     373202    2667424
hda3              2.77         7.37        40.41    1697931    9315298
hda4              0.00         0.00         0.00         10          0
hda5              0.56         3.13         6.99     721189    1610274
hda6              0.00         0.03         0.00       7810        320
hda7              0.00         0.02         0.00       3778        136
hda8              0.00         0.01         0.00       1742        368
hdc               0.00         0.00         0.00        168          0

[root@centos1 ~]#
```

Fig 5 Execute iostat and take a look at its output

Note that the iostat command generates several report lines that can be used to monitor and subsequently change the system configuration to better balance the I/O workload between physical disk devices.

System administrators need to be aware of another very useful command – vmstat. This command helps them in getting information about virtual memory, processes, memory, paging etc. Let us see the output displayed by vmstat in the screenshot below (**Fig 6**).

It will be a good idea to take a look at the detailed man page of vmstat command to understand more about it. If system administrators are interested in collecting data every two seconds, five times, then they can execute this command:

```
$ vmstat -S M 2 5
```

System administrators will get very useful information from vmstat and they will use it to identify various system bottlenecks. Vmstat provides several options. When used with the '-a' option, it gives information about active and inactive memory pages.

What all information vmstat can provide?

Process – 'r' indicates the number of processes waiting for run time and 'b' gives the number of processes in uninterruptable sleep mode

Memory – gives information about free space, memory used as cache etc

Swap – details about swap related fields

I/O – gives information about i/o

System – one can get the system related information such as the number of interrupts per second.

CPU – gives various percentages of CPU time

Another useful command that system administrators need to know is pmap. This command is used to report a memory map of a process or processes. This command can be of significant help in finding out the causes of memory bottlenecks. We take a look at its usage in **Fig 7**.

If we are interested in displaying process memory information for a specific PID (process ID – a unique identification for a process), then we can use the following syntax:

```
[root@centos1 ~]# pmap -d  PID
```

We have used pmap for a specific PID on our system and captured the output in the screenshot (**Fig 8**).

The last line of the output looks like this...

```
mapped: 4668K    writeable/private: 436K    shared: 28K
```

It gives three important statistics:

a. The information about the amount of private address space (indicated by writeable/private).

b. The amount of address space this process is sharing with others (indicated by shared).

c. The amount of memory mapped to the files (indicated by mapped).

We can note that pmap will display the complete memory usage of every command, file and library associated with the process.

```
[root@centos1 ~]# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0    184  24764  45492  66828    0    0     6    29  125  120  0  1 98  0  0
[root@centos1 ~]#
```

Fig 6 A look at vmstat output

```
[root@centos1 ~]# pmap
Usage: pmap [-x | -d] [-q] pid...
-x  show details
-d  show offset and device number
-q  quiet; less header/footer info
-V  show the version number
[root@centos1 ~]#
```

Fig 7 How to use pmap

```
[root@centos1 ~]# ps
  PID TTY          TIME CMD
10927 pts/2    00:00:00 bash
11786 pts/2    00:00:00 ps
[root@centos1 ~]# pmap -d  10927
10927:   -bash
Address   Kbytes Mode  Offset           Device    Mapping
00307000       4 r-x-- 0000000000307000 000:00000   [ anon ]
00427000      36 r-x-- 0000000000000000 003:00002 libnss_files-2.5.
00430000       4 r-x-- 0000000000008000 003:00002 libnss_files-2.5.
00431000       4 rwx-- 0000000000009000 003:00002 libnss_files-2.5.
0052c000     104 r-x-- 0000000000000000 003:00002 ld-2.5.so
00546000       4 r-x-- 0000000000019000 003:00002 ld-2.5.so
00547000       4 rwx-- 000000000001a000 003:00002 ld-2.5.so
0054f000    1276 r-x-- 0000000000000000 003:00002 libc-2.5.so
0068e000       8 r-x-- 000000000013f000 003:00002 libc-2.5.so
00690000       4 rwx-- 0000000000141000 003:00002 libc-2.5.so
00691000      12 rwx-- 0000000000691000 000:00000   [ anon ]
00696000       8 r-x-- 0000000000000000 003:00002 libdl-2.5.so
00698000       4 r-x-- 0000000000001000 003:00002 libdl-2.5.so
00699000       4 rwx-- 0000000000002000 003:00002 libdl-2.5.so
0493b000      12 r-x-- 0000000000000000 003:00002 libtermcap.so.2.0
0493e000       4 rwx-- 0000000000002000 003:00002 libtermcap.so.2.0
08047000     696 r-x-- 0000000000000000 003:00002 bash
```

Fig 8 A look at pmap details for a specific process

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             Stat
e
tcp        0     52 ::ffff:129.221.8.204:ssh    inblr-panigrsk.eu.uis.:4966 ESTA
BLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ]         DGRAM                    1133   @/org/kernel/udev/udevd
unix  2      [ ]         DGRAM                    5599   @/org/freedesktop/hal/u
dev_event
```

Fig 9 Details provided by the netstat command

```
[root@centos1 ~]# strace
usage: strace [-dffhiqrtttTvVxx] [-a column] [-e expr] ... [-o file]
              [-p pid] ... [-s strsize] [-u username] [-E var=val] ...
              [command [arg ...]]
   or: strace -c [-e expr] ... [-O overhead] [-S sortby] [-E var=val] ...
              [command [arg ...]]
-c -- count time, calls, and errors for each syscall and report summary
-f -- follow forks, -ff -- with output into separate files
-F -- attempt to follow vforks, -h -- print help message
-i -- print instruction pointer at time of syscall
-q -- suppress messages about attaching, detaching, etc.
-r -- print relative timestamp, -t -- absolute timestamp, -tt -- with usecs
-T -- print time spent in each syscall, -V -- print version
-v -- verbose mode: print unabbreviated argv, stat, termio[s], etc. args
-x -- print non-ascii strings in hex, -xx -- print all strings in hex
-a column -- alignment COLUMN for printing syscall results (default 40)
-e expr -- a qualifying expression: option=[!]all or option=[!]val1[,val2]
   options: trace, abbrev, verbose, raw, signal, read, or write
-o file -- send trace output to FILE instead of stderr
-O overhead -- set overhead for tracing syscalls to OVERHEAD usecs
-p pid -- trace process with process id PID, may be repeated
-s strsize -- limit length of print strings to STRSIZE chars (default 32)
-S sortby -- sort syscall counts by: time, calls, name, nothing (default ti
-u username -- run command as username handling setuid and/or setgid
```

Fig 10 Usage and options with strace

This is a good way to find out exactly what may be causing high memory consumption in a runaway process and whether or not it is something you can fix.

There is always a need to understand network statistics (such as network connections, interface statistics etc). Linux provides a very powerful command – netstat – for this purpose. Let us take a look at the information displayed by executing this netstat command (**Fig 9**).

Linux also provides another command – ss – that shows information similar to netstat. One can use the ss command to obtain socket statistics.

It is very important for system administrators to understand another command – known as strace. This command is used to trace system calls and signals. Strace is of great help while debugging web server and other server problems. This command shows how data is passed between the program and the kernel. Take a look at the screenshot (**Fig 10**) to understand its usage and various available options.

The beauty of strace is that when it is run in conjunction with a program, it outputs all the calls made to the kernel by the program. In many cases, a program may fail because it is unable to open a file or because of insufficient memory. And tracing the output of the program will clearly show the cause of either problem. Running strace is pretty straightforward. It generally takes the following format:

```
$ strace <program name>
```

Please note that strace also allows you to attach processes for just-in-time debugging. Suppose a process seems to be spending a lot of time doing nothing. A quick way to find out what is going on is to type 'strace -c -p my_pid' at the command prompt.

Server monitoring is considered to be an important and critical activity. In the Linux environment, various commands and utilities are available that will help the administrators to monitor the server. Various solutions have come up in the market and they make server monitoring a relatively easy task. If system administrators have a good knowledge of various commands (as described in this article), they will be able to take care of their servers in an efficient way.

The web browser displaying the various PHP settings

In this file you can also see the various PHP modules you have installed and their settings

Using the phpinfo function of PHP, you can get a lot of information about your server and web server

Turn your Ubuntu system into a LAMP development server

Convert your Ubuntu desktop into a powerful LAMP development server by configuring tools like Apache, MySQL, and PHP

A LAMP server is a server consisting of Linux, Apache, MySQL, and PHP. This is a very popular combination used for web applications. Before we go about setting up the LAMP stack, make sure that you have a computer with Ubuntu Linux installed on it. In this tutorial we will build a LAMP server for your development use. This means that this server will be used locally, as opposed to a server that is live on the internet. We will see how to install and test tools such as Apache, PHP, MySQL and some other useful tools that will make the development of your web application easier.

01 Install Ubuntu and updates

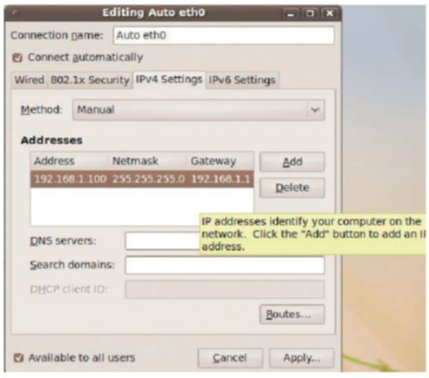
The first step you need to perform is to install Ubuntu Linux on your computer, if it's not already there. We're using Ubuntu Linux 9.10 for this exercise, but the instructions should work on other releases as well. After the installation is complete, set up the internet connection and make sure that you are able to surf the web on your computer. Once that is sorted, you should update all the packages on your computer. This will make sure that we will use the latest versions of packages along with their updates and patches. To do this, go to System>

Administrator>Update Manager. Hit the Check button to get the latest info and then the Install Updates button to complete the process.

Install all the updates for your computer to make sure you have the latest version of the software

02 Post-install config

Once the installation is complete, we need to make two small changes. The first is to ensure that the machine has a static IP address as opposed to a dynamic one. Launch the Network Connections tool from the System>Preferences panel and set this up. Give your computer an IP address that is not already part of the dynamic range of IPs in your DHCP server to avoid conflicts with other computers. The IP will be something like 192.168.1.100.

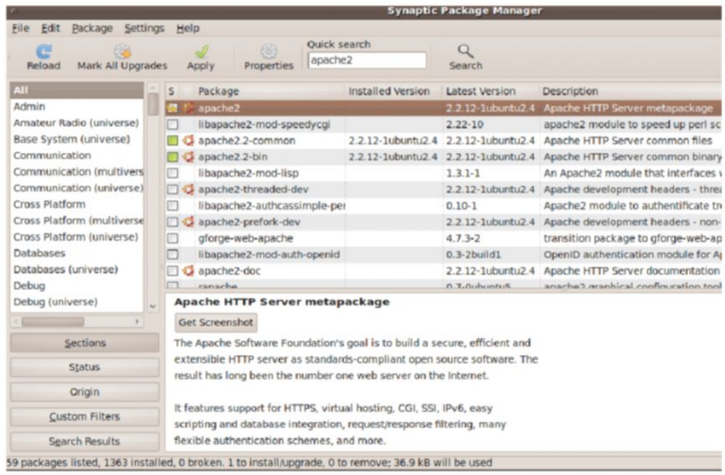


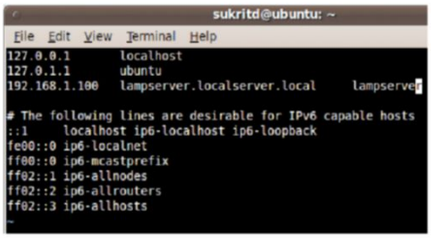
The network settings

03 Set hostname

The other setting change you need to make is to change the hostname of the machine. Open the hostname file with the command '#sudo vi /etc/hosts' and then make a new entry like '192.168.1.100 lampserver.localserver.local lampserver'. You can change the hostname to something you like. Also, execute the command '# sudo echo lampserver.localserver.local > /etc/hostname' to set the hostname. Reboot your machine for the settings to kick in. Now you can check if your new settings worked by executing the command '# hostname'. You should see the new hostname in the output.

Fig 1 Install Apache The installation of the apache2 package





Configure the /etc/hosts file before you set up the hostname

04 Install Apache

We now have Linux installed with all the latest updates. We can proceed to the next step: installing the Apache web server. Launch the Synaptic Package Manager and search for the package 'apache2' and install it along with all its dependencies (**Fig 1**). You can also execute the command '# sudo apt-get install apache2' from the Terminal window if you like.

05 Test web server

Once Apache is installed, it should be started automatically. If not, you can start it manually by executing the command '# sudo /etc/init.d/apache2 start' (**Fig 2**). Once the web server has started, launch your favourite web browser and enter http://127.0.0.1 in the address bar and hit Enter. You should see a page that says 'It works!'. If you see this, your web server has been set up correctly.

06 Installing PHP

PHP is a server-side scripting language that needs to be installed and configured with Apache. If you were to install PHP from source, you would have to perform several steps to configure it with Apache. However, the Ubuntu team has packaged PHP in such a way that all you need to do is to install the package; the rest is taken care of. Execute the command '# sudo apt-get install libapache2-mod-php5 php5' (**Fig 3**, overleaf), or search for the packages 'php5' and 'libapache2-mod-php5' in Synaptic and install them.

07 Test PHP

Once the installation of PHP and its dependencies is complete, you should run a quick test to check if it was installed correctly. It's a good idea to restart your web server at this point in time. Use the command '# sudo /etc/init.d/apache2 restart' for that. Now create the file /

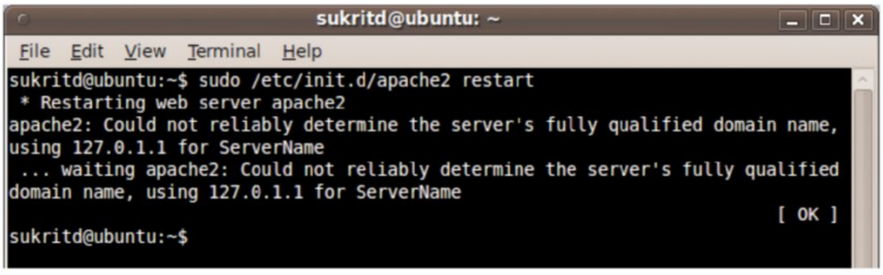


Fig 2 Test web server Start your web server

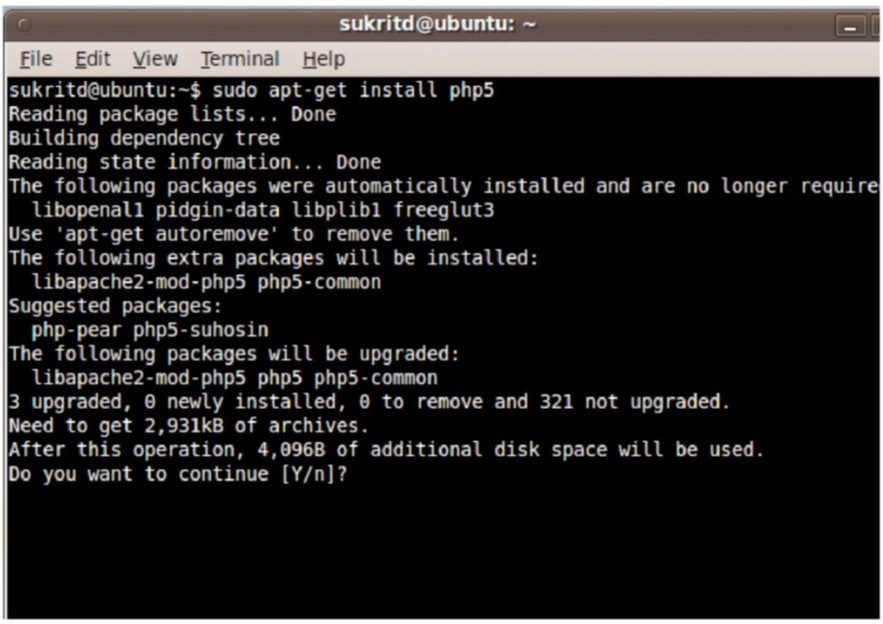


Fig 3 Installing PHP When you install PHP you also need to install some other packages

var/www/phpinfo.php. Execute the command '# sudo vim /var/www/phpinfo.php' to do so. Insert the following line of code into it: <?php phpinfo(); ?>. Launch your web browser and go to http://localhost/phpinfo.php to run the test.

08 MySQL

With Apache and PHP installed, configured and tested, we can move onto the database. We'll be using the MySQL database server. We will need to install the MySQL server. Execute the command '# sudo apt-get install mysql-server' to install it, or search for 'mysql-server' in Synaptic (**Fig 4**).

09 Set up MySQL root password

Whether you installed the MySQL server using Synaptic or the Terminal window, you will be asked to make some configuration during the installation. You will need to tell the installer what the root password for the MySQL server should be. You can skip this step, although we strongly urge you to have a strong password.

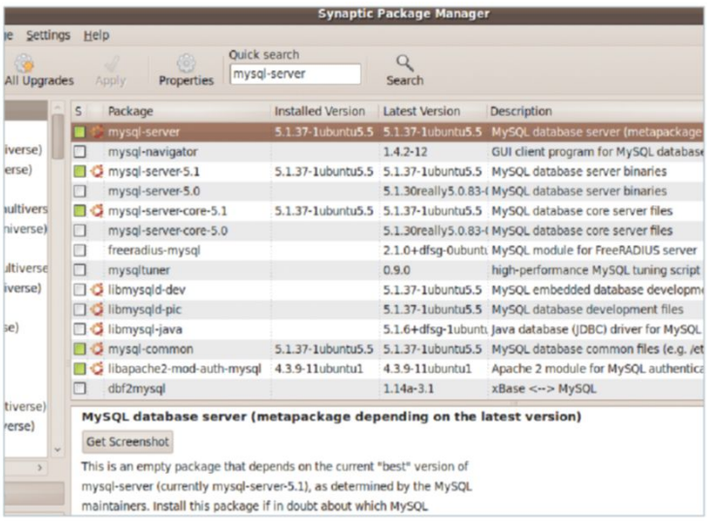
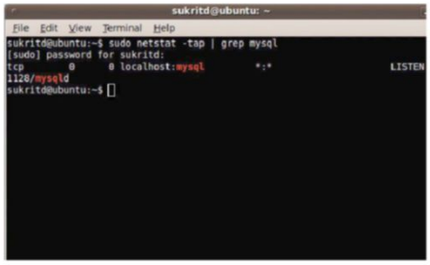


Fig 4 MySQL The MySQL server installation

10 Test MySQL

Execute the command '# sudo netstat -tap | grep mysql' to quickly check if the MySQL server is running. If you see an output like in the screenshot below, the server is running. Now log in to make sure you got the root password right. If you execute the command '# mysql -u root', you should get an error saying that you are denied access. The correct way to log into MySQL would be to use the command '# mysql -u root -p'. You will now be prompted to enter the password of the root user. Enter it and you should see the MySQL prompt. Voila, MySQL is all ready.



Check the network ports to make sure that the MySQL server is running

11 Create a MySQL database and user

Using MySQL as root in your applications is not a very good idea. So let's create a database and a user for you to use in your applications. Log into MySQL as the root user. Now execute the following commands in the MySQL Terminal. Create a new database: 'mysql> CREATE DATABASE worlddomination;'. Now create a new user and grant him/her some limited privileges for this database: 'mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES ON worlddomination.* TO 'stewiegriffin'@'localhost' IDENTIFIED BY 'lois1s3vil';'. Enter a last command before quitting MySQL: 'mysql> flush privileges;'. You can read up about granting MySQL privileges so you can figure out what your app needs and grant privileges accordingly. Now you should be able to log in with the new MySQL username and password.

Advisor

Sukrit Dhandhania has spent several years working professionally, implementing several open source tools for companies. During this time he has evaluated, set up and maintained various open source tools for these firms

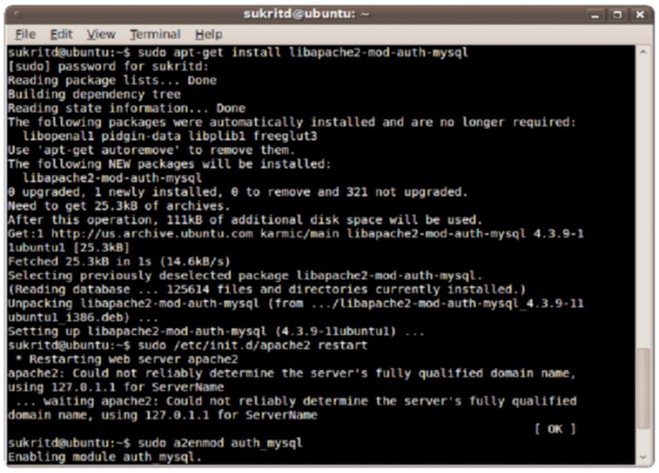


Fig 5 Connect MySQL with PHP Set up one last package to sync MySQL with PHP

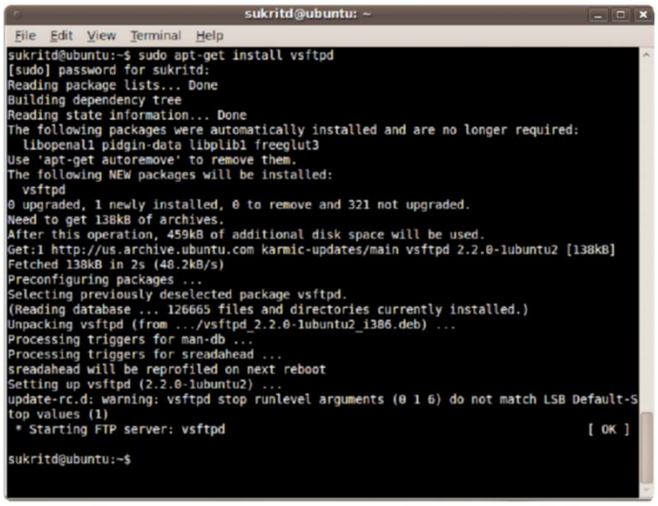
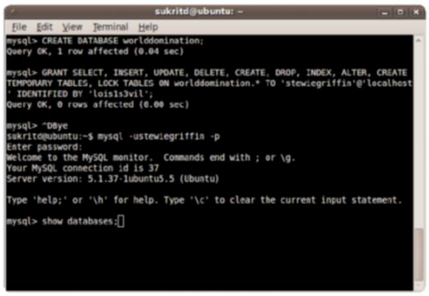


Fig 6 Installing an FTP server The vsftpd installation process



Create a database for your application, along with a user. Grant only the necessary privileges to the user

12 Connect MySQL with PHP

MySQL needs to be integrated with your PHP installation. There is a module that does this for you. Install it using the command '# sudo apt-get install libapache2-mod-auth-mysql'. Enable the installed module in PHP using the command '# sudo a2enmod auth_mysql' (**Fig 5**). Restart your web server. Now launch your browser and access the phpinfo.php file again. Search for the term 'mysql' on the page and you should find that the module has now been activated.

13 Install phpMyAdmin

phpMyAdmin is a wonderful PHP web application that makes managing your database a breeze. Install the package with Synaptic or the command line using '# sudo apt-get install phpmyadmin'. You need to make a few small configuration settings during the installation. After the installation completes, you should then be able to go to http://localhost/phpmyadmin and log into your database server and view and make changes to your databases.

The phpMyAdmin login screen

14 Installing an FTP server

You now have a functional LAMP server installed. You can begin developing your web applications and testing them out. You may also need an FTP server to upload your files if you are not using SCP to do so. There are certain advantages that the limitations of FTP bring, such as restricting the user to a directory. We'll use the vsftpd server for this. Install it with '# sudo apt-get install vsftpd' (**Fig 6**).

15 Setup vsftpd

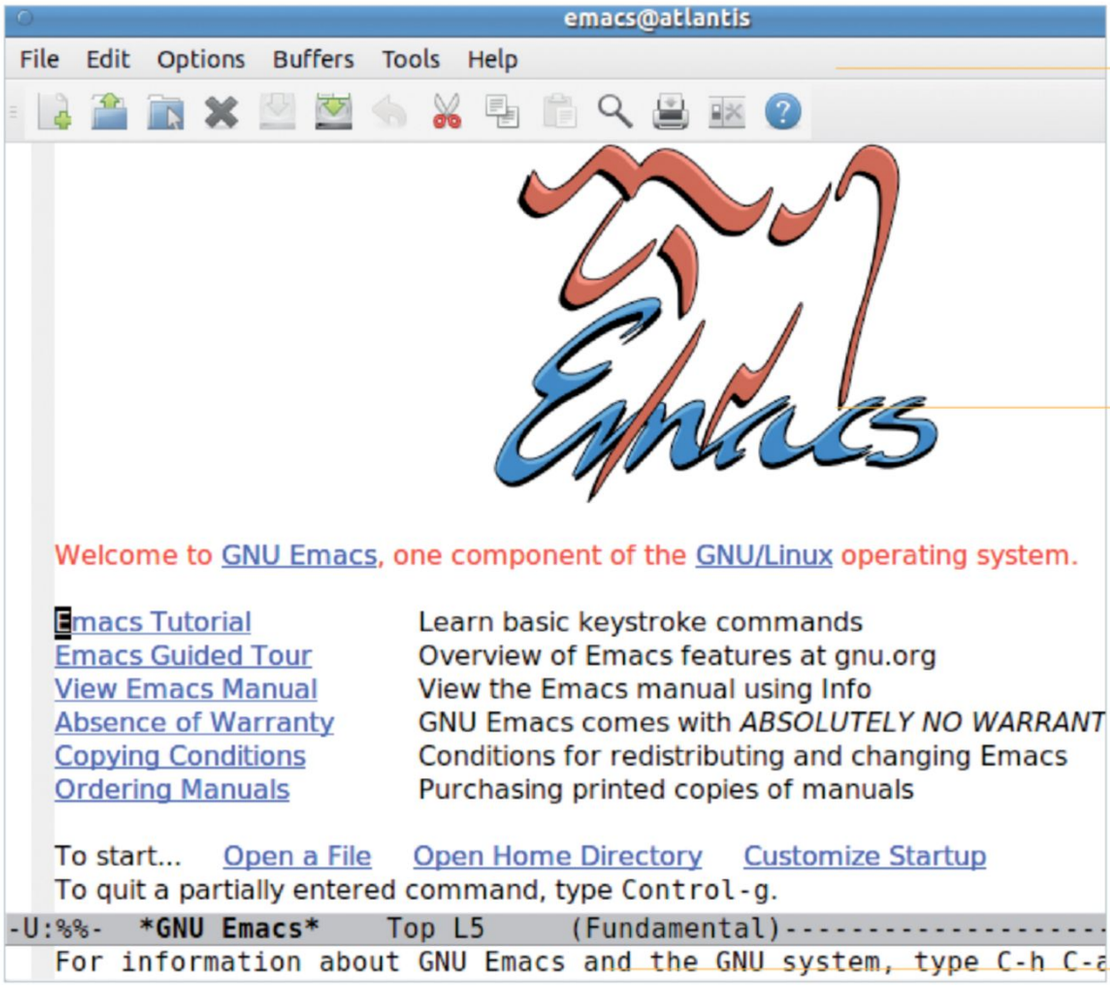
Create a folder for the contents of your FTP server to be stored. We used the directory /opt/ftp for this purpose, created with '# sudo mkdir /opt/ftp'. Now change the home folder of the ftp user to the new folder using the command '# sudo usermod -d /opt/ftp ftp'. Finally, restart the FTP server with '# sudo /etc/init.d/vsftpd restart' to activate it. Now you can copy the files and folder you want to use in your FTP setup to this location.

16 Enable vsftpd security

Open the vsftpd configuration file with '# sudo vi /etc/vsftpd.conf'. Make the following changes to the settings: 'local_enable=YES', 'write_enable=YES', and 'chroot_local_user=YES'. Save the file and restart the FTP server – '# sudo /etc/init.d/vsftpd restart'. The FTP server is now all set up. You can log in using your username and password. Create more users if you like.

17 Additional packages

There are some other packages that you can install to make your development process a bit easier. Some of these packages are openssh-server, which is a tool that allows you to create a secure terminal connection to your LAMP server; vim-nox, which is a version of the Vim text editor; and SVN, which is a very useful tool that allows you to store various versions of your code files. Another important thing you should set up is backups. You should either write a script or use a ready-made tool to help you take regular backups.



This is the main menu bar for Emacs. It changes depending on which mode you are in. All of the commands are also accessible through Ctrl key combinations

This is the main buffer. It can be broken down into subsections while editing multiple files, or getting output from commands

This is the main modeline for Emacs. It displays information about Emacs and the file you are editing

Build a 'Hello World' app with Emacs

Emacs is the old stalwart of programmer's editors. In this tutorial we will build a GTK 'Hello World' application and learn how Emacs can make development work easier

Advisor

Joey Bernard's day job as an HPC research consultant has given him lots of different development experience. Joey has been using Linux since version 2.0 of Slackware

Emacs (originally short for Editor MACroS) started in 1976 as a set of macros for the editor TECO (Text Editor and COrrector). It has been called the best operating system masquerading as an editor. It is now up to version 23, with version 23.2 having been released on 8 May 2010. It runs on Linux, the various BSDs, Solaris, Mac OS X, Windows, DOS and many other platforms. There is support for editing lots of different programming languages, you can control subprocesses, and there is the ability to interact with lots of different version control systems, such as CVS and Subversion. Emacs is also extensible, through LISP modules. There are a huge number available online, and you are free to

Resources

Emacs for your OS
http://www.gnu.org/software/emacs/

CEDET http://cedet.sourceforge.net

Splint http://www.splint.org

A compiler (such as GCC)
http://gcc.gnu.org

write your own to add almost any functionality that you can think of. We'll look at writing a 'Hello World' program to introduce the tools and techniques available through Emacs to help you in your development work. By the end of this article, you should have all the information you need to go off and develop your own application the old-school way.

01 What is Emacs?

There are two main camps in the editor religious wars: vi and Emacs. Vi is a modal editor, where you have a command mode and an edit mode. Emacs only has one mode. Commands are executed by using key combinations. These key combinations invariably start with either the Ctrl key or the meta key (this is usually assigned to the Alt key), or some combination of them. If you are used to vi, you will have to do a little brain retraining, but we think it will be worth it. Note that in the rest of this tutorial, 'C-' will represent the Ctrl key, while 'M-' will represent the meta key (**Fig 1**).

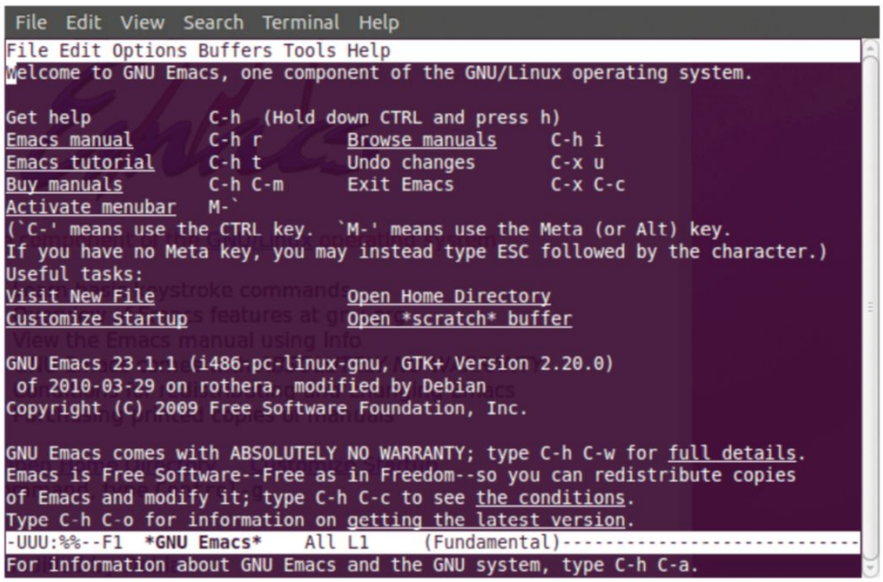


Fig 1 Emacs welcome screen with some keyboard shortcuts

02 Starting a new project

Out of the box, Emacs doesn't have a lot of support for project creation or management. The usual workflow is to create files and directories by hand, either at the command line or from within Emacs. To create a new file in the current directory, you would use 'C-x C-f filename', where filename is a file which doesn't exist yet; Emacs will create this new file for you. To change what Emacs considers is the current directory, you would type 'M-x cd', and Emacs will prompt you for the directory name. If the directory doesn't exist yet, you will have to create it with 'M-! mkdir dirname'.

There is a project, CEDET, which provides several tools to help with software project management. If your needs are greater than this tutorial can satisfy, you should check it out.

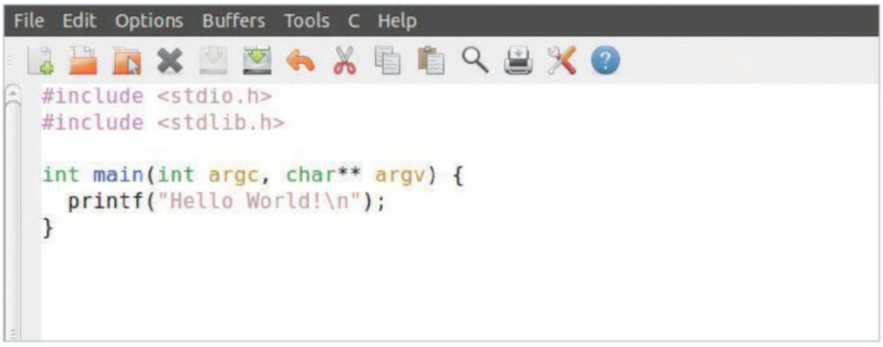
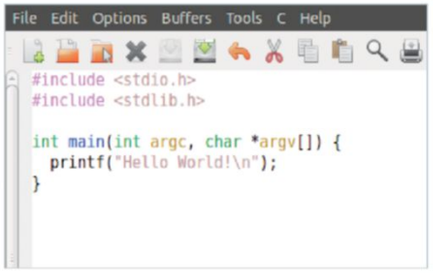


Fig 2 Editing code and managing files



> **"It has been called the best operating system masquerading as an editor. It is now up to version 23, with version 23.2 having been released in May 2010"**

03 Managing files

Emacs has lots of file management facilities built in. To load a file into the editor, you would use 'C-x C-f'. Emacs will then ask you for the filename. You are then free to edit the source code and get some actual programming done (**Fig 2**). When you're done editing, you can save those changes with 'C-x C-s'. If you have multiple files open at the same time, they occupy

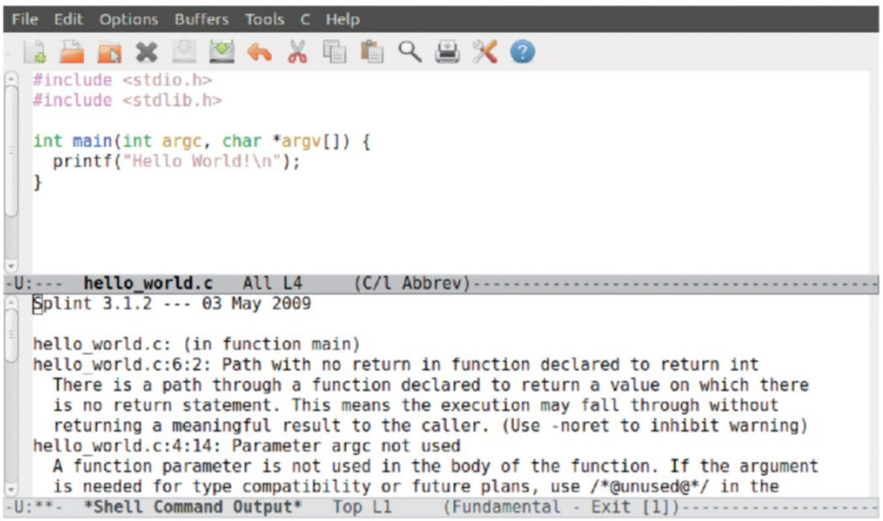
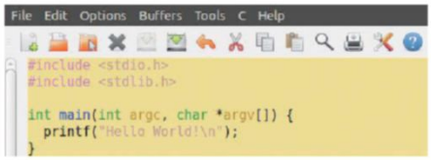


Fig 3 Checking your code with Splint

different buffers. To move from buffer to buffer, you use 'C-x b'. By default, Emacs will offer to change to the last buffer you were at, but you can select a different one if you have several open.

04 Code formatting

Emacs tries to do smart code formatting for you, based on the language you are programming in. Since we are looking at C, Emacs tries to do indenting the way you expect to see it. It also highlights matching parentheses, so that you can do a visual inspection. To indent a single line, you simply have to hit Tab on that line. If you wish to format an entire region, you would use 'C-M-\'. So, to format the entire file, you would select the entire buffer with 'C-x h' and then use 'C-M-\' to do the formatting.



05 Checking your code

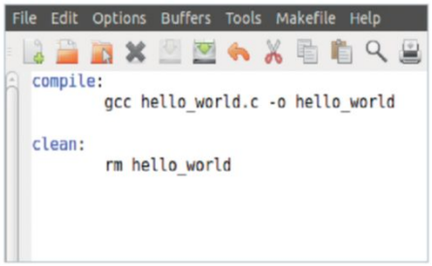
Emacs doesn't have the ability to check your code directly for errors, but there is a utility, called Splint, which can do this. When you have your program all finished, you can run Splint against it to check for errors. This is probably a good thing to do, just as an initial test of code correctness. To do this, you would use 'M-!' to start a shell subprocess, then run 'splint hello_world.c' and the output will be displayed in its own buffer (**Fig 3**). Remember to take the results with a grain of salt, as no tool is perfect.

06 Compiling your code

Emacs expects that you are using makefiles in your project. When you are ready to compile, you need to type in 'M-x compile'. Emacs will generate a command of the form 'make -k' (**Fig 4**). You can add any extra options here, or if you wish, you can enter an explicit compilation command like 'gcc hello_world.c -o hello_world'. When you do the compile, a new buffer is created to display the output. If you run into an error, you can get Emacs to jump to the relevant line in the source file by typing 'C-x `'. This will go to the location of the first error. Typing it again will take you to the next error location, and so on.)

07 Makefiles

Since Emacs assumes that you are using Make and makefiles, you can use Emacs to edit the makefile. Targets in the makefile are denoted by a name starting in column one and ending with a colon. The lines following this are the commands to run for this particular target. These commands must start with a Tab character.



> **Emacs doesn't have the ability to check your code directly for errors, but there is a utility, called Splint, that can do this**

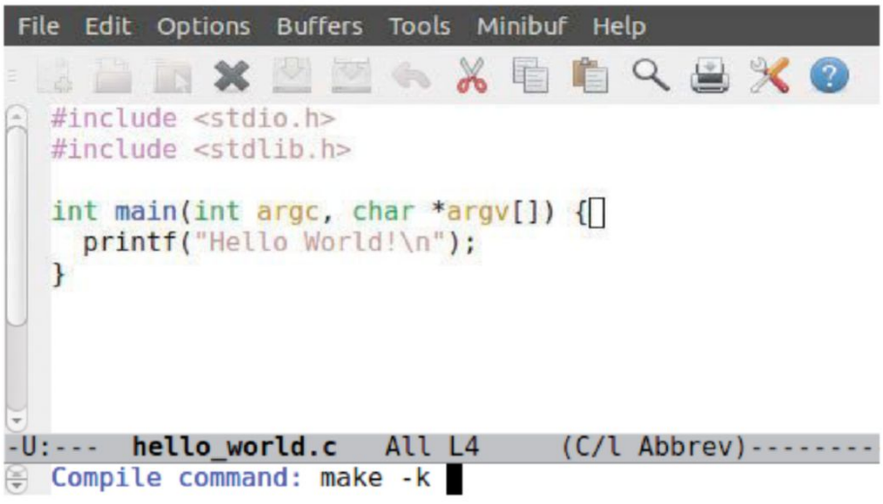


Fig 4 Compiling your code

> Emacs has built-in support for debugging with GDB. To start the debugger, type 'M-x gdb' and hit Enter. Emacs will give you a GDB command line

08 Debugging code

Emacs has built-in support for debugging with GDB, the GNU debugger. To start the debugger, simply type 'M-x gdb' and hit Enter. Emacs will then give you a GDB command line which you can edit to remove or include any options you need. Then you hit Enter to start GDB. Emacs actually has a GDB mode, so the menu bar has additional entries for things you can do with GDB (**Fig 5**).

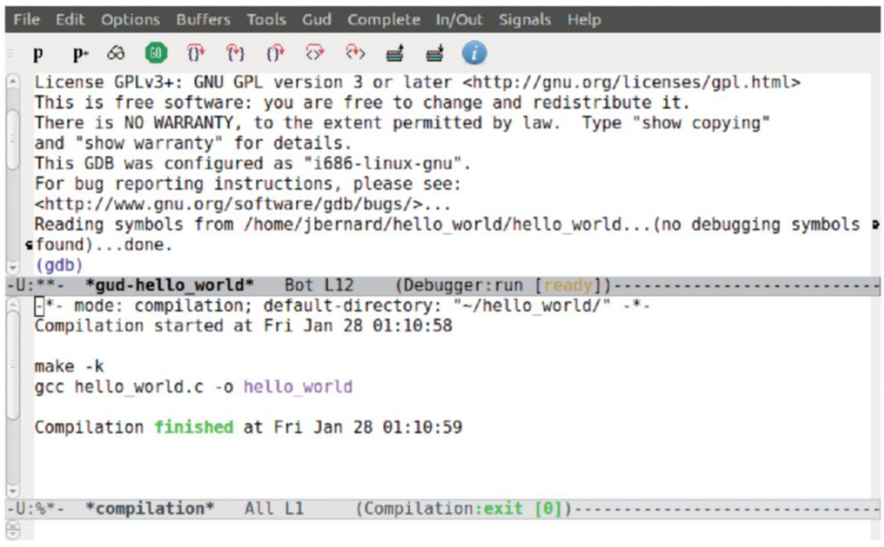


Fig 5 Debugging code with GDB

09 Version control

Emacs has built-in support for several version control systems, including Arch, CVS, Git, RCS and Subversion. When you are editing a file that is under version control, details about this are displayed in the modeline (**Fig 6**). There is a generic super-command which executes the next most logical step when using version control. This command is 'C-x v v'. For example, if you just edited a file that was in CVS, hitting 'C-x v v' would check in those changes. If you don't want to accept what Emacs wants to do with your file, you can prefix the command with 'C-u', ending up with 'C-u C-x v v'. Emacs will then let you make changes to the command before executing it.

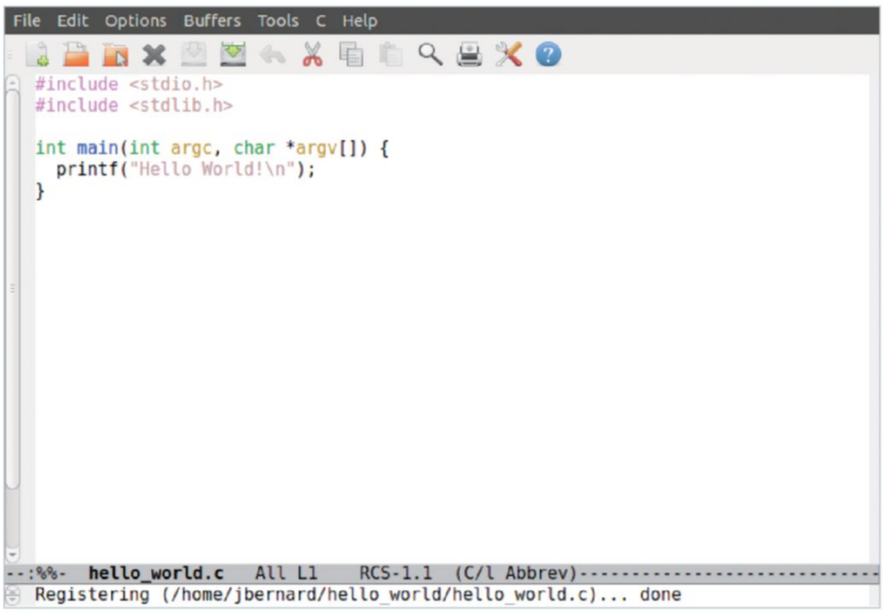


Fig 6 Version control

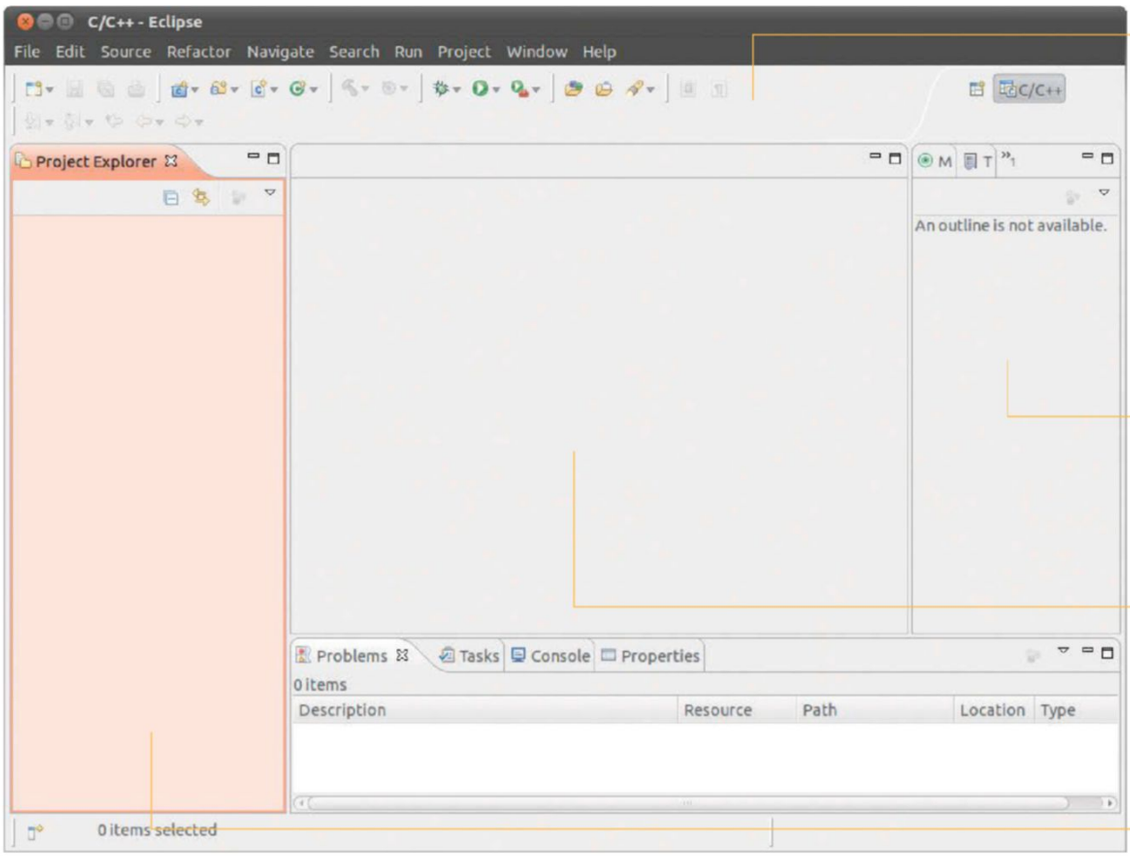
> Emacs has built-in support for several version control systems, including Arch, CVS, Git, RCS and Subversion

10 Final thoughts

This article forms only the most basic introduction to using Emacs as a development environment. It should get you up and running with your software projects. But each of these steps could be a tutorial in itself, so don't be afraid to look at the documentation and continue the learning process. And never be afraid to ask questions.

Developing software with the Eclipse IDE

We present a crash course in developing software with Eclipse, the fully featured IDE written in Java



> **Support for several different programming languages, as well as different source code control systems, is available through plug-ins that can be downloaded and installed**

Eclipse is both a project and a foundation. The software project was started in November 2001 by IBM. The Eclipse Foundation was created in January 2004 to steward the Eclipse community. Because there are GUI components, it only runs under Windows, Linux and Mac OS X. Eclipse is built around a plug-in architecture. Support for several different programming languages, as well as different source code control systems, is available through plug-ins that can be downloaded and installed. There are

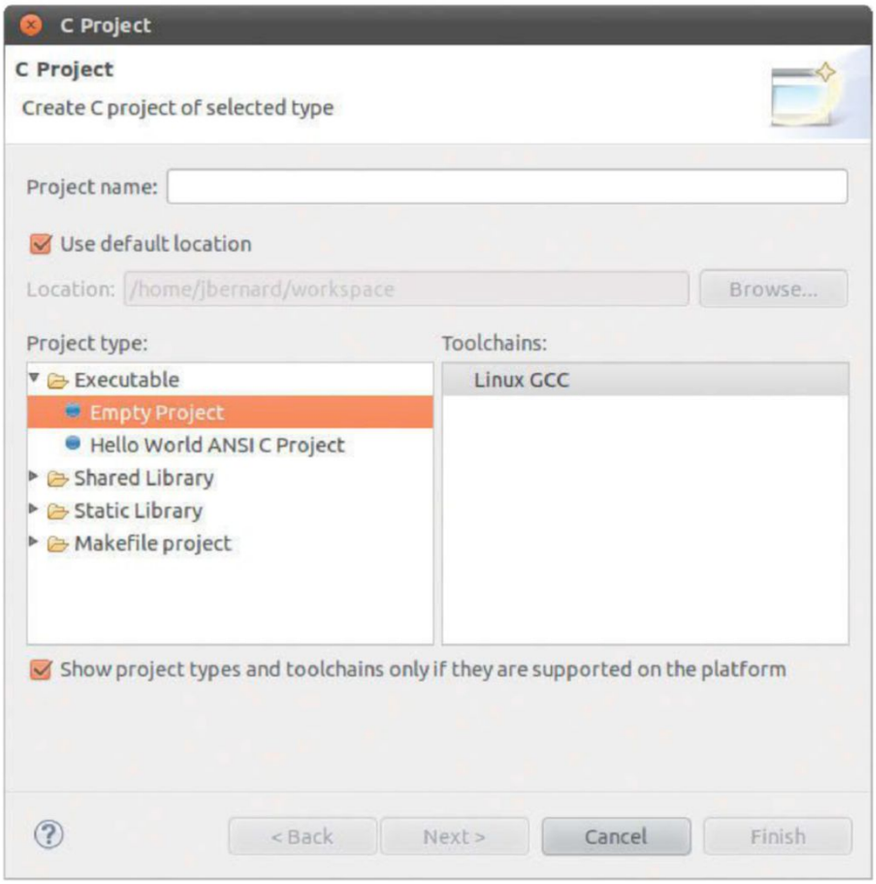


Fig 1 Starting a new project

also plug-ins to tie into bug tracking systems, handle communications, and even connect to other machines. Since Eclipse plug-ins are written in Eclipse, you have all the tools you need to write your own plug-ins. We'll look at writing a 'Hello World' program to introduce the tools and techniques available through Eclipse to help you in your development work. By the end of this article, you should have all the information you need to go off and develop your own application in a nice GUI IDE.

Advisor

Joey Bernard's day job as an HPC research consultant has given him lots of different development experience. Joey has been using Linux since version 2.0 of Slackware.

1. What is Eclipse?

Eclipse started out life as an IDE for Java development. The original design philosophy was around having a framework which you could add plug-ins to. Very rapidly, plug-ins were written to support other languages, such as C and PHP. When enough interest gathered around a group of functionality, a new project was created – in this case, CDT (C/C++ Development Toolkit). Eclipse is a fully GUI IDE (integrated development environment), so warm up your mousing hand. But, like all applications, there are key combination shortcuts that you can learn and use.

2. Starting a new project

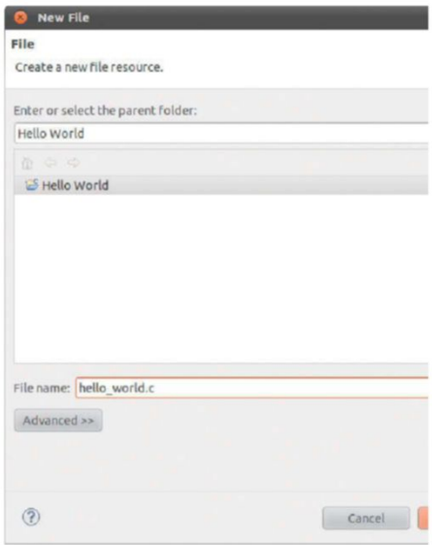
Eclipse, by itself, doesn't have much support for C programming projects. The CDT download of Eclipse includes the ability to create several different types of projects. To create a new C project, click on File>New>C Project. This will open a new window where you can select either an empty project, or a basic 'Hello World' project (**Fig 1**). You can pick which toolchain you want to use; the default that most people will use is GCC. You can also choose to create a project to make a shared library or a static library. You do have to give a name to your project (at the top of the window), and you can choose whether to put this project in the usual workspace directory or off in another directory.

Resources

Eclipse CDT for your OS
http://www.eclipse.org/cdt/

Java for your OS http://java.sun.com

A compiler (such as GCC) http://gcc.gnu.org



3. Managing files

In Eclipse, you have two choices in dealing with files. You can use the menu bar at the top of the screen. You can create new files, delete files and do all kinds of operations. Equivalently, you can simply right-click on the project that is listed in the left-hand pane (called the Project Explorer) to create a new file. You can then right-click on a file to delete it, rename it, copy it or open it in an editor. You can have multiple files open at once. They show up listed as tabs in the editor window. This concept should be familiar to anyone who uses a modern web browser.

4. Code formatting

Eclipse will try to do some automatic code formatting for you as you are writing your code. It will automatically add ending brackets and parentheses, as well as ending quote

marks. When you open a new code block (defined with curly braces) and hit Enter, Eclipse will automatically indent the next line of code. If, while editing, the indentation levels get messed up, you can force Eclipse to re-indent a line by placing the cursor on the line in question and clicking Source>Correct Indentation, or right-clicking and selecting the same options (**Fig 2**). There is also a key-combo to do the same (Ctrl+I). If you want to correct an entire region, simply highlight the area in question with your mouse before issuing the command.

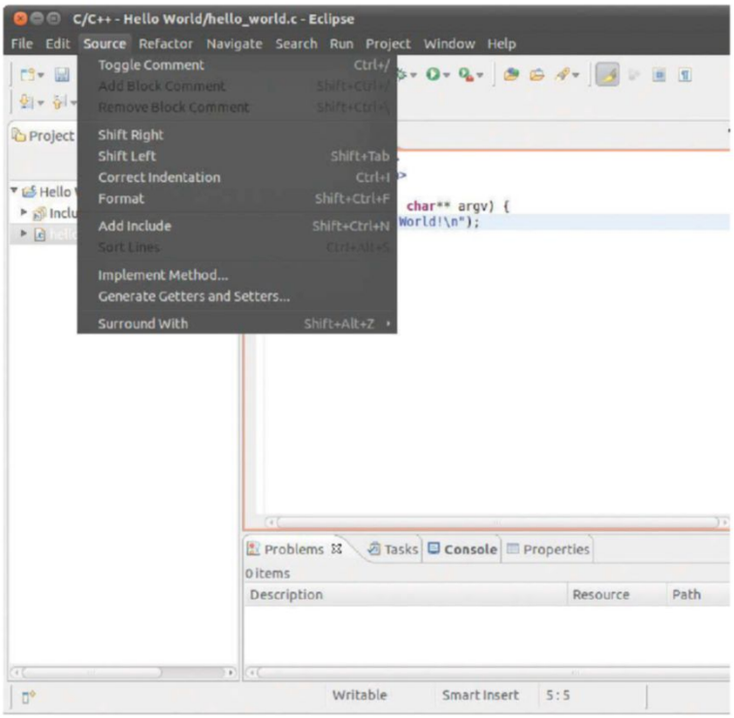
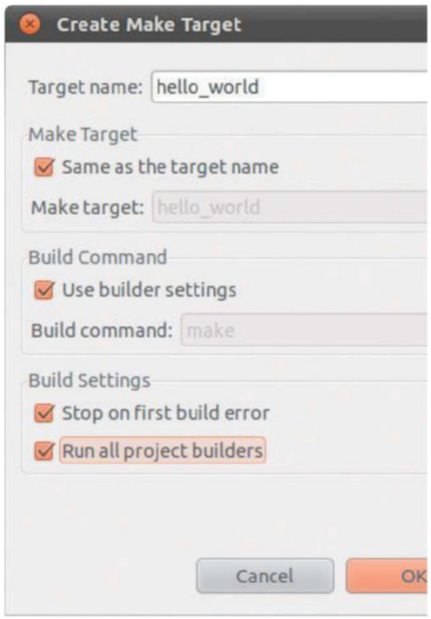


Fig 2 Correcting indentation

5. Checking your code

Eclipse doesn't have the ability to check your code directly for errors, but there is a utility, called Splint, which can do this. When you have your program all finished, you can run Splint against it to check for errors. This is probably a good thing to do, just as an initial test of code correctness. To do this, you would open a new Terminal window, run 'splint hello_world.c' and then check the output. Remember to take the results with a grain of salt, however, as no tool is perfect.



6. Compiling your code

Eclipse has two types of projects: Eclipse ones or makefile-based ones. If you created a basic C project at the beginning of this process, then you have an Eclipse build process setup. If you have a single source file project (like our small example), building is as simple as clicking Project>Build Project. You can also right-click on the project in the left-hand pane and select Build Project. If you run into any errors, they will show up in the bottom pane. As an example, if we forget the semi-colon on the printf line, we get the error seen below (**Fig 3**). You can right-click on the error in the Problems window and it will take you to the problem line in the source.

7. Makefiles

Eclipse can handle makefiles, but instead of editing them directly, you can manage them through the 'Make Target' view in the right-hand pane. Select that tab from the list to bring it to the front. You can select the New Make Target

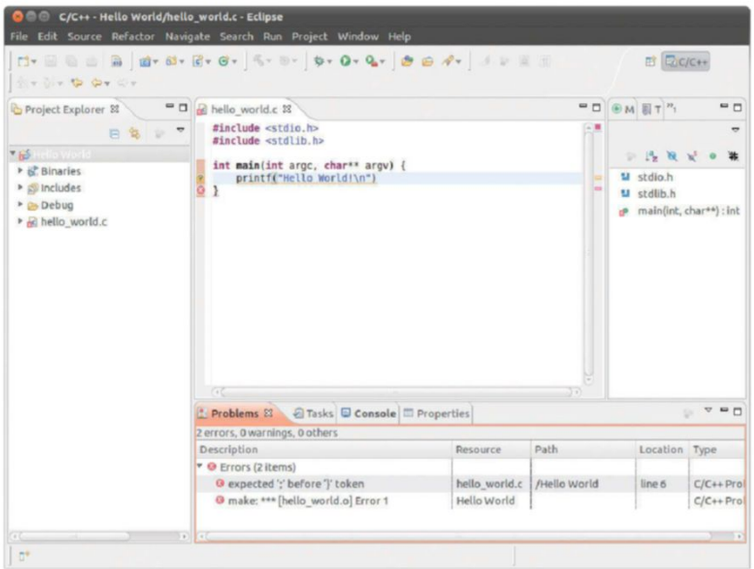


Fig 3 Errors are shown in the bottom pane

> **Eclipse has support to tie into GDB, providing a wonderful GUI interface for the GNU debugger**

button, or right-click on the top level and select 'New...'. This will create a new target in the project makefile. You can always check out the actual makefile in your project directory if you want to do any hand edits.

8. Debugging code

Eclipse has support to tie into GDB. When you want to run your code under a debugger, you simply need to click on Run>Debug. If you have more than one target that can be built, you can select which one to debug. After you select one, the main window changes over to the debug perspective, which provides a nice GUI interface compared to the horror that is GDB on the command line. Anyone who has used a commercial debugger, like Totalview, will see very familiar information on these panes (**Fig 4**).

9. Version control

Eclipse has built-in support to handle source code management through CVS. If you want to use some other system, such as Git or Subversion, you will need to install a plug-in for it. If you want to use the default CVS, you access it by right-clicking on your project in the left-hand pane. Go down to Team and click on 'Share Project...'. This will open a new window where you can set the details for your CVS server (**Fig 5**). Eclipse supports ext, extssh, pserver and pserverssh2 as the connection methods. You can set the host and repository path. There are equivalent settings windows for the other source code management systems.

xxxxxxxxxxxxxx

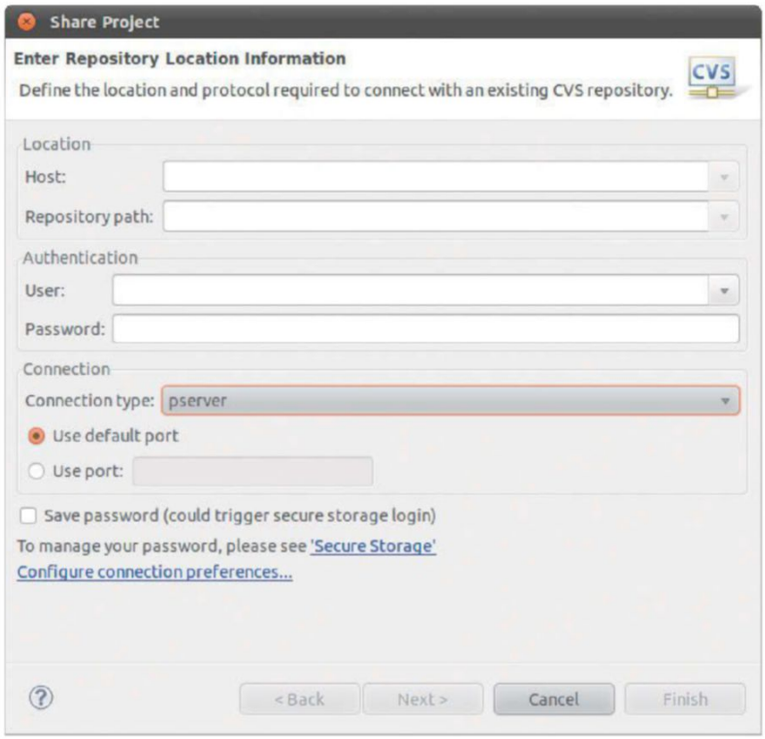


Fig 5 Enter CVS server details

10. Final thoughts

This article is only a basic introduction to using Eclipse as a development environment. To add other plug-ins from the main repository, click on 'Help>Install New Software...' to make a new window appear, where you can choose (**Fig 6**). You can also enter another repository site to load even more plug-ins. And as always, you have all the tools you need to write your own included in Eclipse. Have fun and hopefully more productive programming.

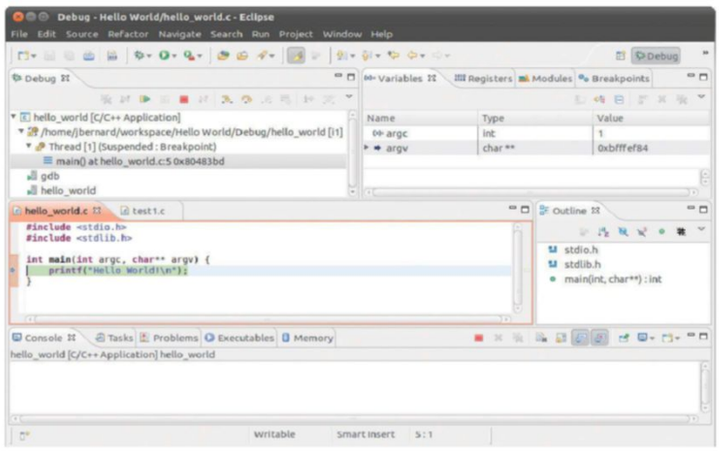


Fig 4 Debugging your code

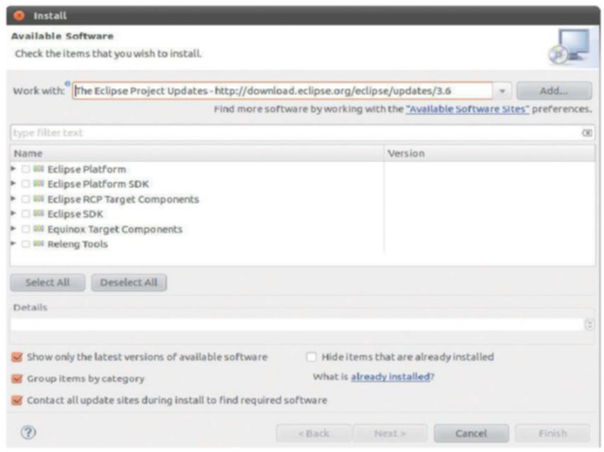


Fig 6 Download more plug-ins

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Debugging masterclass

Save the relationship between you and your application and don't be left hanging with 'Segmentation fault' ever again...

To err is human (or to be a software developer). If not for those errors there would be no software updates or even a QA (quality assurance) department and we'd all have hair on our heads. Nonetheless, life is not perfect and neither is software. That is just fact. Our goal as software developers is to minimise the bugs and make sure that users have a nice experience using our software. Nobody likes to use a calculator which explodes and burns into ashes when dividing a number with zero. (This is a calculator used for crucial process in an aeroplane. Got the exploding part now?)

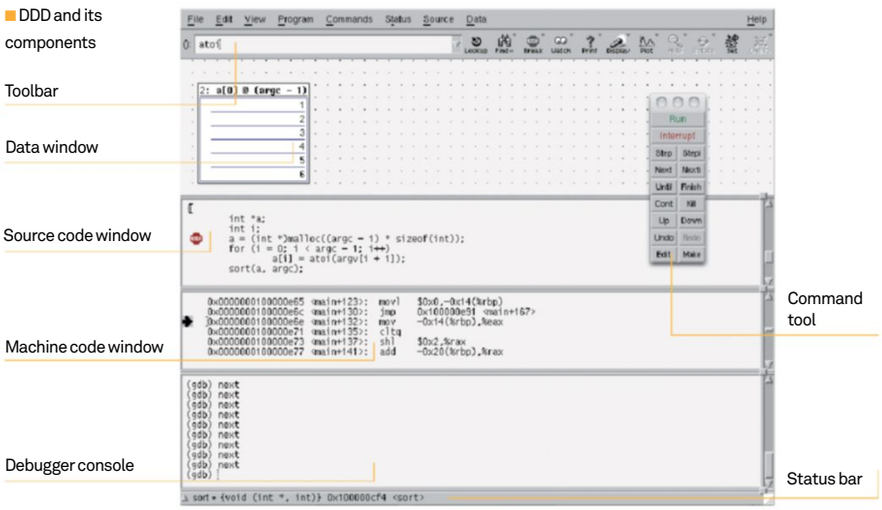
We usually don't go into much detail about how a particular term originated. But we are making an exception for 'debugging' for obvious reasons. 'Debugging' was coined by Rear Admiral Grace Hopper. She was working on a Mark II computer at Harvard University; her associates discovered a moth stuck in a relay and thereby impeding operation, whereupon she remarked that they were 'debugging' the system.

> **"Our goal as software developers is to minimise the bugs and make sure that users have a nice experience using our software"**

Common sense: no tools required

Like they say about deadly diseases, prevention is better than cure. You can get rid of most of your problems if you just stay alert in the beginning. There is no point in running loops that you cannot end and cause the calculator to explode. The following are a few tips that you should keep in mind to prevent you falling off the proverbial cliff...

1. Compiler warnings: Most compilers, including our beloved GCC, do not show all the warnings by default. But warnings are important, even the ones you cannot see by default. Most of the warnings have a very good chance of turning into potential bugs. This situation gets more critical when you are working using a dynamic language such as Objective C, which will happily compile a message that does not actually exist (it will warn about it though) and then the app will



crash if this message is called during the execution of program.

There are two things you must do. Use the GCC option -Wall to enable all the warnings associated with the code and then use -Werror to treat those warnings as errors. This means you won't be able to compile your program until you get rid of the warnings.

2. Defensive programming: Defensive coding is like reserving a space in an underground shelter to save yourself in case of a nuclear fallout. Defensive coding practices are used to prevent or reduce errors in case the program is misused. In our calculator example, a defensive coding practice would to check for division by zero rather than discovering this problem after shipping the software. Another important example of defensive coding is making your software safe from all the known security vulnerabilities. This is not to say that terrorists will be eyeing your program, but it is just to make sure that users are protected from at least all known vulnerabilities.

The following two code snippets describe examples of defensive coding…

@code:Noob Code

```
switch (Orientation)
{
   case Orientation.Horizontal:
      break;

   case Orientation.Vertical:
      break;
}
```

@code: Defensive Code

```
switch (Orientation)
{
    case Orientation.Horizontal:
        break;
    case Orientation.Vertical:
        break;
    default:
        break;
}
```

In the above snippet, if the program cannot receive the Orientation value either as Horizontal or Vertical, it will do something rather than crashing.

3. Comment your code: Many developers think of themselves as geniuses when they are not. Actually, they are often so dumb that they cannot understand their own code, let alone by some other developer. That's why we have comments. They not only allow the original developer to understand the system, but also help any other developers who will be joining the development team. This helps to prevent a lot of unnecessary bugs early on.

4. Unit testing: Out of all our common-sense tips, this the biggest no-brainer. Unit testing is a method of writing tests for each of your units (usually functions) to check the behaviour of the functions when the inputs are of various types (correct, erroneous, upper/lower limits etc). To do this, you do not have wait for your program to finish; rather, do it as often as possible. In fact, you should write tests as you are developing the project (also known as test-driven development). If you do this, you will have two benefits: your programs will have fewer bugs and you can call yourself an extreme programmer – no kidding, since unit testing is the cornerstone of extreme programming, which relies on an automated unit testing framework.

GDB with DDD: welcome to the jungle

GDB (GNU Debugger) is the default debugger on Linux systems (also on most of the systems not made by Microsoft, such as Mac OS X, BSD etc). It is a very robust and very capable debugger. However, it operates only in text mode. That's why a newbie or anyone coming from a Windows graphical environment might find it dull-looking and difficult to use. Moreover, the command-line interface is not well suited for large systems with complex data structures.

So we have DDD (Data Display Debugger), which is a very popular graphical user interface for GDB. DDD supports a number of languages, including Ada, Bash, C, C++, Chill, Fortran, Java, Modula, Pascal, Perl and Python. It can also be used with other popular debuggers such as XDB (on HP UX systems), DBX (found on other UNIX flavours), JDB (Java Debugger), PYDB (Python Debugger), Perl Debugger and even Bash debugger (**http://bashdb.sourceforge.net**).

DDD provides all the features of GDB (since it acts as a front end to GDB). But this doesn't set limitations for DDD. It takes full advantage of its graphical user interface. **DDD provides a number of features to its users, including:**

1. It can display data structures in a graphical format and is able show relations between them graphically.
2. View source texts and breakpoints.
3. Hypertext source navigation and lookup, ie the ability to switch between multiple source files automatically.
4. Machine-level debugging.
5. Breakpoint, back trace and history editors.
6. Change program attributes, such as variables and function arguments, in runtime.
7. The same interface is used for all languages supported by DDD.

Note

At this point you should install DDD in your system, because we will jumping on to some hands-on stuff very soon. Search your distro's package manager to look for it.

DDD user interface

Let's take a look at the important parts of the DDD application...

Toolbar: It provides options for entering arguments and running common tasks which are associated with a given argument, like displaying data structures, rotating displays, setting breakpoints etc. An argument can be entered in the argument field labelled ():.

Data window: It shows data related to the program being debugged. It may contain array diagrams and other displays.

Source code window and machine code window: Self-explanatory.

Debugger console: The console shows GDB's (or the debugger you are using) command prompt. This accepts classic debugger commands. It also shows debugger commands, output and messages.

Command tool: It provides buttons for the most used commands.

Debugging with DDD

In this section we will quickly start using DDD. To do this we will go through a process to debug our test program, shell.c. Following is the source code for shell.c. This program is expected to sort given integers. The shell sort algorithm is used to implement sorting. (Read that data structure book again for more information on shell sorting.)

> **"After a few seconds you'll be up and running... Now we'll execute our program step by step, keeping an eye on what's happening during execution"**

> **"In this section we'll start using DDD. To do this we will go through a process to debug our test program, shell.c"**

@code:shell.c

```
#include <stdio.h>
#include <stdlib.h>
static void sort(int a[], int size)
{
   int i, j;
   int h = 1;
   do {
        h = h * 3 + 1;
   } while (h <= size);
   do {
        h /= 3;
        for (i = h; i < size; i++)
        {
                int v = a[i];
                for (j = i; j >= h
&& a[j - h] > v; j -= h)
                        a[j] = a[j
- h];
                if (i != j)
                        a[j] = v;
        }
   } while (h != 1);
}

int main(int argc, char *argv[])
{
   int *a;
   int i;
   a = (int *)malloc((argc - 1) *
sizeof(int));
   for (i = 0; i < argc - 1; i++)
        a[i] = atoi(argv[i + 1]);
   sort(a, argc);
   for (i = 0; i < argc - 1; i++)
        printf("%d ", a[i]);
   printf("\n");
   free(a);
   return 0;
}
```

Compile this program using the following command:

```
$ gcc -g -o shell shell.c
```

Note: We are not doing a normal compilation. Normally binary programs contain only information that is needed to serve its purpose to keep the file size to a minimum. Hence it doesn't have any debugging symbols. To incorporate debugging you need to use the -g option, which tells the compiler to include debugging symbols with the executable.

Let's execute the code to see how it is doing:

```
$ ./shell 2 3 4 5 1
1 2 3 4 5
```

That's perfect!! Let's try one more...

```
$ ./shell 2 3 4 5 1 6
0 1 2 3 4 5
```

Note

Output may vary depending upon the compiler used. For example, the Apple build of GCC (4.2.1) on Mac OS X will exhibit this bug in both cases. This code is compiled using GCC 4.4.3 (Ubuntu).

This second result is not good. Here, 6 is not shown and a new number 0 is shown. There is something wrong with this code. Let's use DDD to find what exactly is going on inside this program.

Start DDD with our program

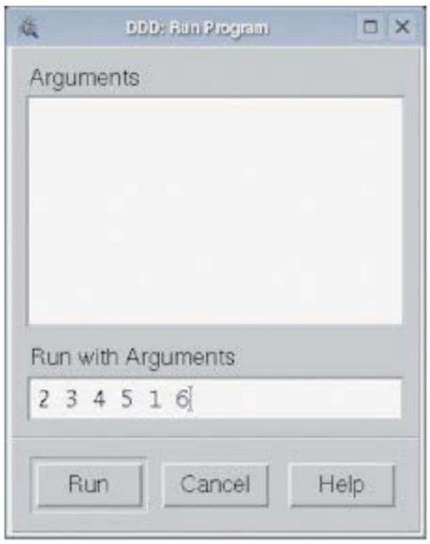
```
$ ddd shell
```

After a few seconds you'll be up and running with your DDD session. Now we'll execute our program step by step, keeping an eye on what's happening during execution. The first thing we need to do is set a 'breakpoint' in order to prevent the program finishing in one go. A breakpoint causes suspension of program execution whenever a certain point is reached and gives us the opportunity to interact with our debugging system and perform actions.

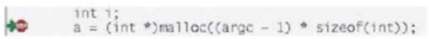
We will set a breakpoint on initialisation of 'a'. To do this, click on the empty space before 'a', where it is being initalised (line number 27), then click on the Break button in the toolbar to create a breakpoint. The argument field should read now 'shell.c:27'. You will see a red stop sign on line 27.

Now it's time to run the program. To do so, select Program from the menu bar and then click

Run... The Run Program dialog will be presented to you. In the 'run with arguments' box, enter the arguments which caused wrong output '2 3 4 5 1 6' and click Run to start the execution.



The current execution line is indicated by a green arrow.



Now we will need to navigate through the code. There are two major ways of stepping through source code. One is Next, which takes you line by line and skips the function call (and presents result of call), and the other is Step, which will go through function calls (subroutine calls) as well. We will use both commands as required.

You can also look inside the current values of variables. To find the value of a variable, move the mouse pointer over it and wait for a second: a small window with the value pops up. You can also see values at the status bar.

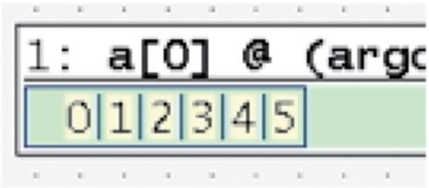
You can view the contents of an array graphically. To view individual values, you can type the array with its index number (eg a[0]) in the argument field found in the toolbar and then click the Display button. To display all members of an array, you must use the @ operator. For example, to view all elements of 'a', enter 'a[0]@(argc-1)' in the argument field and then click Display. (argc-1) is something on which values of a [ ] is dependent.

If the display is vertical, you can click the Rotate button to show it horizontally.

To proceed further in this program, click the Next button a number of times and you will see the values of array a [0] changing. Changed numbers are highlighted in the display.

To resume execution of the loop, use the Until button. It is used to run the program until a line greater than the current one is reached. Click on the Until button until you reach the end of the call to the sort() function.

At this point, the contents of the array should be as follows:



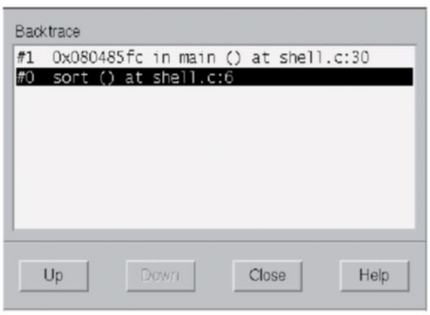
Now we will investigate the cause behind this. Clear the previous breakpoint by clicking on the old breakpoint and then selecting the Clear button (on toolbar). Create a new breakpoint on the sort function call (line number 30). Run the program again using Program>Run Again. **To see what's going on with the sort function, click the Step button until the current line points to:**

```
=>int h=1;
```

Debugger console shows the current function

```
(gdb) step
Sort (a=0x804b008, size=7) at shell.
c:6
```

To see the whole function call stack, you can use Status>Backtrace from the menu bar.

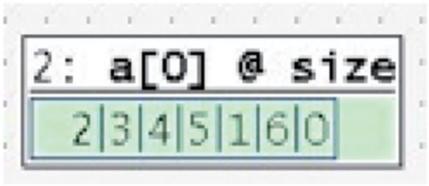


Now we will check whether the argument passed to the sort function was correct or not. For this we will have to see values for array a[ ], now in the context of the current function.

For this, enter a[0]@size in the argument field and then click Display.

So we have:

> **"With DDD, people who are new to programming can easily debug their programs"**



So are you wondering where the last 0 comes in? Actually it is a bogus value that comes due to the wrong value of 'size'. That is what causing the problem, a wrong parameter. To verify this we will set the correct value of size. To do this, click on size and then click on the Set button. Set the value for size equal to 6. To set the value of size, click the variable size then click Set on the toolbar. Click OK and then click Finish to resume execution of the sort function.

Hurrah! Now you can see the correct values for a[ ].

So now you know what was causing the problem. We can now fix the source code.

Change

```
sort(a, argc); to
sort(a, argc-1);
```

Compile and run again to check if everything is correct now.

This was a quick introduction to DDD, a friendly GDB interface. DDD has lots more to offer, from linked data display to graphs and more. With this tool, people who are new to programming (as well as those from other GUI environments) can easily debug their programs. Since DDD provides a single interface for all the programming languages it supports, it'll be easier to debug any code without worrying about the complexities of the individual underlying debugger. For more details, take a look through the man pages.

Debugging networks with traceroute

Swayam Prakasha talks us through traceroute, the popular network debugging utility...

Traceroute is a very popular network debugging utility. It measures the route path and transit times of packets across an IP (internet protocol) network. This utility is available on all operating systems. It is considered one of the powerful tools and it helps the network administrator to find out a wealth of data about their network and the internet. It is typically used to find the potential bottlenecks between your computer and a remote computer across the network. In simpler terms, traceroute gives a listing of all the router hops between your server and the target server. This in turn helps you to verify whether routing over the networks in between is correct. It can help you to determine why your connections to a given server might be poor, and can often help you figure out where exactly the problem is.

One can consider traceroute as a command-line tool used for testing and validating routing functionality. If a trace completes on the destination machine, then we are sure that

Advisor

Swayam Prakasha has

a master's degree in computer engineering. He has been working in information technology for several years, concentrating on areas such as operating systems, networking, network security, electronic commerce, internet services, LDAP and web servers. He can be reached at swayam.prakasha@gmail.com

How traceroute works?

Traceroute works by increasing the TTL value of each successive batch of packets sent. The first three packets sent will have a TTL value of 1, anticipating that they are not forwarded by the first router. The next three packets have a TTL value of 2, so that the second router will send the error reply. This continues until the destination host receives the packets and returns an ICMP Echo Reply message.

The traceroute utility uses the returning ICMP messages to produce a list of hosts that the packets have traversed in transit to the destination. The three timestamp values returned for each host along the path are the delay (or the latency) values, typically measured in milliseconds for each packet in the batch.

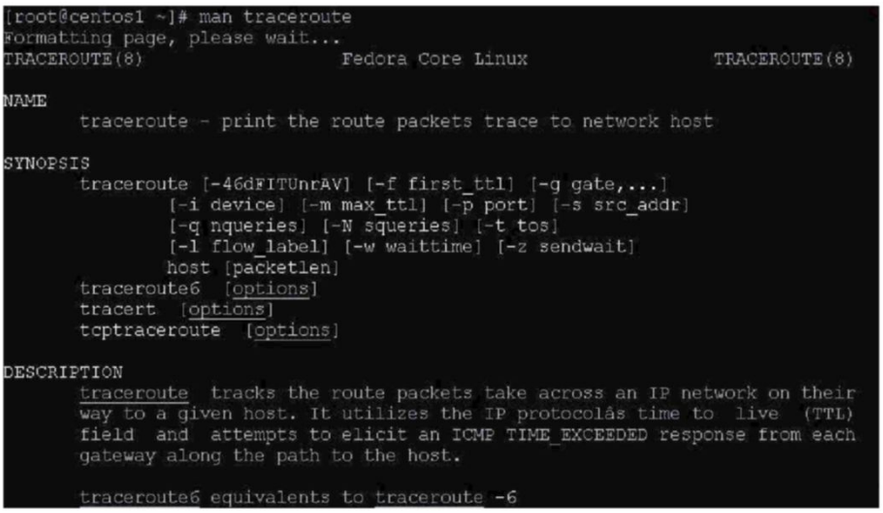


Fig 1 The very detailed man page for traceroute

there is a route to the destination. Traceroute can provide the user with information about the hosts through which traffic passes. It is important to note here that traceroute will not show all possible paths through a network to a destination, and usually only shows the most preferred path.

It is a classic tool that helps to find the route taken by a packet while travelling from a source to a destination. It uses TTL (time-to-live counter) for its operation. The sender transmits ICMP (Internet Control Message Protocol) echo requests with varying TTL values and listens to echo reply messages. Each router, after processing the datagram, decrements the TTL. When TTL reaches zero, the router discards the datagram and 'an ICMP time exceeded message' is transmitted back. The route is determined by examining these time-exceeded messages sent by intermediate routers. The traceroute program also contains a client interface to ICMP. Like the 'Ping' utility (please refer to the Ping feature in issue 97 of Linux User and Developer), it may be employed by a user to verify that an end-to-end internet path is operational, but also provides information on each of the intermediate systems (ie IP routers) to be found along the IP path from the sender to the receiver. Traceroute uses ICMP echo messages. These are addressed to the target IP address. The sender manipulates the TTL (hop count) value at the IP layer to force each hop in turn to return an error message.

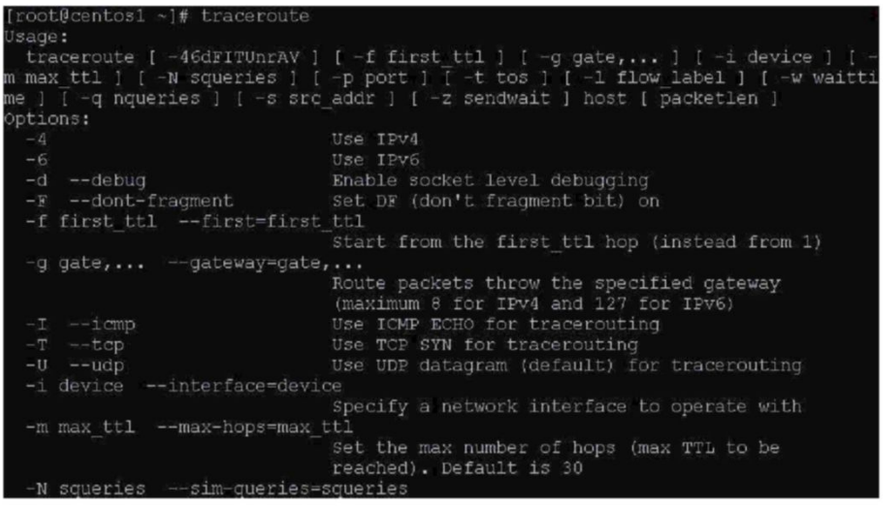


Fig 2 A look at traceroute options

> **"Typically, traceroute is used for troubleshooting networks"**

Typically, traceroute is used for troubleshooting networks. This utility shows a list of routers traversed and this in turn will help a network administrator in identifying the path taken to reach a very specific destination. With this information, one can identify routing problems or firewalls that may be blocking ICMP traffic. Penetration testers also commonly use traceroute so that they can gather information on network infrastructure and IP ranges around a given host.

Traceroute comes with an excellent man page. Readers are advised to go through this man page in detail. This is shown in the screenshot on the left (**Fig 1**).

It is also a good idea to take a look at various options that are available with traceroute. Take a look at the screenshot below (**Fig 2**) to understand these various options.

Let us take a look at the usage of traceroute. **It is pretty simple and it is given below.**

```
[root@centos1 ~]# traceroute
129.221.5.202 (Note that
129.221.5.202 is my machine's IP
address).
```

The output of the above traceroute is shown as follows...

```
traceroute to 129.221.5.202
(129.221.5.202), 30 hops max, 40
byte packets
1  129.221.8.193 (129.221.8.193)
1.199 ms  1.324 ms  1.268 ms
2  172.23.240.229 (172.23.240.229)
0.611 ms  1.080 ms  0.897 ms
3  172.23.240.234 (172.23.240.234)
0.680 ms  0.708 ms  6.150 ms
4  inblr-prakass.my_firm.com
(129.221.5.202)  0.451 ms  0.692 ms
0.539 ms
```

One can use 'which traceroute' to locate this utility on your system. Generally, you need to be a root user to run this utility. But if you are using the full path name, then one can run this as a normal user.Let us understand more about the traceroute utility with an example. From our machine, we tried to run traceroute on www.

hotmail.com and the output was captured in the screenshot on the right (**Fig 3**).

For analysis purposes, here we have the output of the traceroute on www.hotmail.com…

```
[root@centos1 ~]# traceroute www.
hotmail.com
traceroute to www.hotmail.com
(64.4.20.169), 30 hops max, 40 byte
packets
 1  129.221.8.193 (129.221.8.193)
1.140 ms  1.208 ms  0.938 ms
 2  172.23.240.229 (172.23.240.229)
2.060 ms  0.590 ms  0.711 ms
 3  172.23.240.125 (172.23.240.125)
1.163 ms  1.552 ms  1.605 ms
 4  172.22.241.86 (172.22.241.86)
0.583 ms  0.588 ms  1.091 ms
 5  eblgnfga.ui.unisys.com
(172.22.241.90)  0.838 ms  0.688 ms
0.630 ms
 6  eblgcrcx.ui.unisys.com
(172.22.241.98)  2.091 ms  3.269 ms
3.102 ms
 7  115.248.176.54 (115.248.176.54)
3.056 ms  3.216 ms  3.087 ms
 8  220.224.182.250
(220.224.182.250)  3.177 ms  3.734
ms  3.571 ms
 9  115.255.252.222
(115.255.252.222)  27.831 ms  30.351
ms  31.101 ms
10  62.216.147.249 (62.216.147.249)
30.069 ms  29.866 ms  30.070 ms
11  so-7-0-0.0.ejr03.sin001.flagtel.
com (62.216.128.73)  61.389 ms
62.136 ms  6           2.298 ms
12  so-0-2-0.0.pjr02.hkg005.flagtel.
com (85.95.26.125)  989.424 ms
1987.933 ms             1988.739 ms
13  so-0-2-0.0.pjr02.wad001.flagtel.
com (85.95.25.189)  988.670 ms
988.555 ms              1988.106 ms
14  so-7-1-0.0.cjr04.tok002.flagtel.
com (85.95.25.202)  147.876 ms
148.749 ms              148.659 ms
15  so-4-2-0.0.ejr02.pao001.flagtel.
com (62.216.128.62)  959.850 ms
so-3-0-0.0.e           jr02.pao001.
flagtel.com (62.216.128.1)  956.460
 ms so-4-2-0.0.ejr02.pao001.flagt
el.com (62.216.128.62)  955.341 ms
16  peer.flagtel.com
(62.216.145.242)  953.182 ms
919.212 ms  1912.060 ms
17  ge-6-3-0-59.pao-64cb-1b.ntwk.
msn.net (207.46.47.241)  911.624 ms
838.192 ms              1814.191 ms
18  xe-0-0-2-0.bay-16c-1b.ntwk.msn.
net (207.46.40.128)  810.364 ms
994.559 ms              991.673 ms
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
```

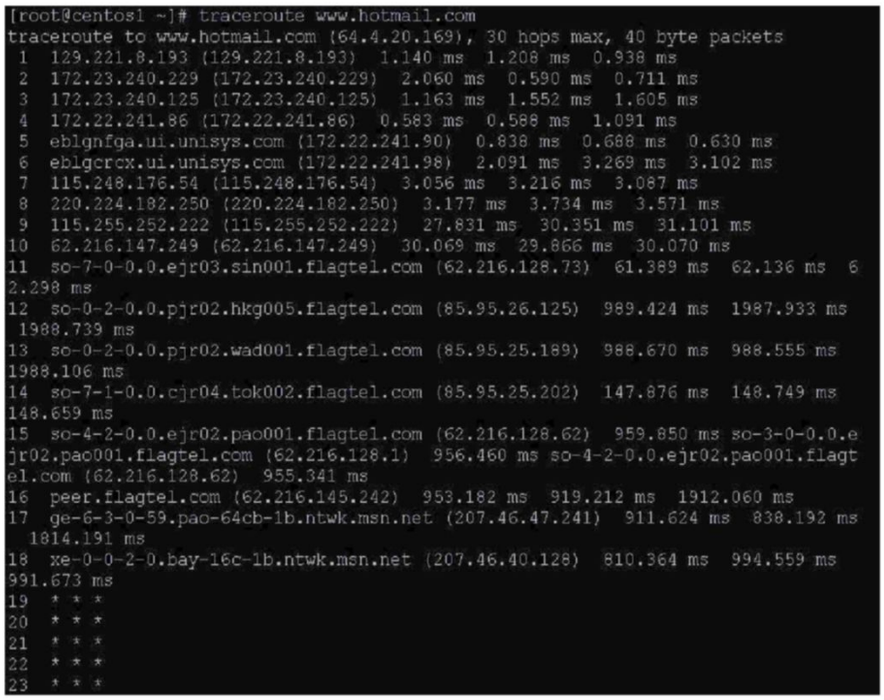


Fig 3 The traceroute output for www.hotmail.com

Table 1 A look at message codes returned by traceroute

***	Time exceeded
!H, !N, or !P	Host, network or protocol unreachable
!X or !A	Communication administratively prohibited. A router Access Control or firewall is in the way
!S	Source route failed. Source routing attempts to force traceroute to use a certain path. Failure might be due to a router security setting

As we can see from the above output, it defaults to a maximum of 30 hops. The first line of the output displays the IP address of the hotmail.com domain. **The other two parameters displayed by the first line are:**

a. The maximum number of hops that traceroute will keep track of the packets before it reaches the destination.

b. The size of the packets, which is 40 bytes.

The following lines show the IP address or domain name of the gateway servers through which the packets pass, as well as the time in milliseconds of the ICMP TIME_EXCEEDED response from each gateway along the path to the host. Basically, we will have one line for each system or router in the path between the source and the target system. On any line, followed by the IP address, we can see three time values in milliseconds. There are three values because traceroute by default sends simultaneously, three packets of 40 bytes each. And the three time values are the time taken to send the packets and receive an ICMP TIME_EXCEEDED response from the gateway. In other words, these three values are the round-trip times of the packets. The round-trip times tell us how long it took a packet to get from us to that system and back again.

In some cases, a line in the output may have

one or two of the times missing, but with an asterisk. This does not indicate a problem, but considered normal. It just means that the system has discarded the packets for some reasons.

In the above output, from the 19th hop onwards, we have received a series of timeouts and this was indicated by asterisks. Thus the point to note here is that timeouts are not necessarily an indication of packet loss. **This timeout issue could be due to several issues and some of them may be…**

a. The network connection between the server on the 19th hop and that on the 20th hop is broken.

b. The server on the 20th hop is down.

It is important to understand the return codes of traceroute. Traceroute can give a number of possible message codes, as shown in the table at the bottom of the preceding page (**Table 1**).

As can be seen from our earlier discussion, whenever there is no response within five seconds timeout, an asterisk (*) is printed.

There is another variation of traceroute, called mtr (Matt's traceroute), and this can be used to do a repeated traceroute in real time. We take a look at its man page in the screenshot at the top right of this page (**Fig 4**).

Let us understand this with an example. We use mtr on the IP 129.221.5.202 and the following screenshot (**Fig 5**) shows its output.

We can note from the above output that the constant updates enable us to visually determine which hops are slow. Matt's traceroute can be used as an effective tool whenever we suspect that there is some intermittent network congestion. One of the important aspects of mtr is that it gives the best, worst and average round-trip times in milliseconds for the probe packets between each hop along the way to the final destination.

Let us see how we can use traceroute output in analysing the network errors.

a. If the trace to a system ends in timeouts (and never completes), then we are sure there is a problem. If the traceroute ends in timeouts at a certain system, it's likely that either the connection between that system and the next system on the route, or the next system itself, is the source of the problem.

```
[root@centos1 ~]# man mtr
MTR(8)                                mtr                                MTR(8)

NAME
       mtr - a network diagnostic tool

SYNOPSIS
       mtr      [-hvrctglspni46]      [--help]      [--version]      [--report]
       [--report-cycles COUNT] [--curses] [--split] [--raw] [--no-dns] [--gtk]
       [--address IP.ADD.RE.SS]   [--interval SECONDS]   [--psize BYTES   |   -s
       BYTES] HOSTNAME [PACKETSIZE]

DESCRIPTION
       mtr combines the functionality of the traceroute and ping programs in a
       single network diagnostic tool.

       As  mtr starts, it investigates the network connection between the host
       mtr runs on and HOSTNAME.  by sending packets with purposly  low  TTLs.
       It  continues to send packets with low TTL, noting the response time of
       the intervening routers.  This allows mtr to print  the  response  per-
       centage and response times of the internet route to HOSTNAME.  A sudden
       increase in packetloss or response time is often an indication of a bad
       (or simply overloaded) link.
```

Fig 4 A detailed look at Matt's traceroute

```
                          My traceroute  [v0.71]
centos1.centos.com (0.0.0.0)                          Mon Feb  7 09:38:26 2011
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                          Packets               Pings
 Host                                   Loss%  Last   Avg  Best  Wrst StDev
 1. 129.221.8.193                        0.0%   2.0   2.1   1.8   2.5   0.3
 2. 172.23.240.229                       0.0%   2.6   2.0   1.4   2.6   0.6
 3. 172.23.240.234                       0.0%   2.0   3.2   1.3   7.4   2.8
 4. inblr-prakass.eu.uis.unisys.com      0.0%   1.8   2.5   1.8   3.5   0.7
```

Fig 5 Output of mtr on a specific IP

b. We can get a round-trip time (or latency time) from traceroute output. Smaller values for latency generally mean better connections. In traceroute output, if you see a significant jump in latency from one hop to another, then that could indicate a problem. In addition to focusing on the root cause here, you can also use the ping program to get a better idea of the latency as well as the packet loss to a given site or router.

In this article we have a taken a look at traceroute, which is considered to be a very useful tool to pinpoint where a network error occurs on the internet. It can also be used to test the responsiveness of a domain or server. For example, If your route to a server is very long (takes over 25 hops), you can expect that the performance will suffer. A long route can be due to less-than-optimal configuration within some network along the way. Also, when you observe the output of traceroute, a large jump in latency (delay) from one hop to the next could indicate a problem.

It is important to note that you will only receive traceroute responses from functioning devices. If a device responds, it is less likely to be the source of your problems.

It is considered best practice to get a bidirectional traceroute – that is not only the traceroute from the source IP to the destination IP, but also from the target IP to the source IP. This is basically because the packet's return path from the target is sometimes not the same as the path taken to get there. Typically, you want to run a traceroute from both ends, but this is not always possible.

> **"You will only receive traceroute responses from functioning devices"**

Debugging your Android applications

Learn how to detect most problems with your Android apps so your users don't have to...

Resources

Sun / Oracle JDK Most Linux systems come with OpenJDK, but it isn't supported very well. You will need to uninstall OpenJDK and install Sun JDK.

Android SDK You can download the Android SDK from http://developer.android.com/sdk/index.html . After installing the SDK, install Android SDK 2.2 or higher and Android SDK Platform-tools. You will also need to add the path <AndroidSDK>/tools and <AndroidSDK>/platform-tools to the path.

Eclipse (3.5 or higher) and Android Development Tools plug-in Both of the tools together make a capable Android IDE (although, if you want something familiar, look at the next item. MOTODEV Studio.) Download the tools from http://developer.android.com/sdk/eclipse-adt.html

MOTODEV Studio for Android (recommended) If you want a more integrated experience, you can install MOTODEV Studio for Android. It packs everything which is included by Eclipse + ADT plus much more, such as automatic Android SDK installation, screen capture, application validator and so on. You can download it from http://developer.motorola.com/docstools/motodevstudio/

An Android device (recommended) The Android emulator is insanely slow (thanks to the native x86 to ARM binary translation). Having a real Android device can speed up things significantly.

Sample code projects here are built using Android SDK 2.3.3 SDK. However, they can also be run on earlier devices. Make sure you edit AndroidManifest.xml and the default.properties file accordingly if you are going to run the code on Android devices with less than 2.3.3 installed.

While we were originally going to title this guide 'build bug-free Android apps', it's not something that's humanly possible. Ask around – any true developer will tell you that a bug-free application is simply an unobtainable goal. If that were possible, you would never see software updates and only major versions would be released. What is possible, however, is to make sure that you debug (or test) your applications thoroughly before releasing them to the general public. Problems are bound to slip through, however, so you'll need to make sure that you listen to your users and do your level best to fix the bugs that they discover. Ultimately, the idea is more to make sure you fix bugs as you find them – this way you get to focus on putting your creativity first, without spending an inordinate amount of time being paranoid about the bugs.

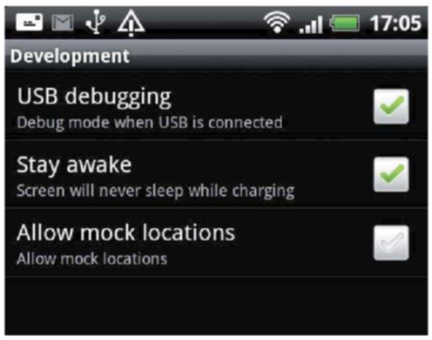
The tools

An engineer is as good as his or her tools. Knowing the right set of tools helps you to stay productive and avoid unnecessary labour. In this section we will look at the tools available to an Android developer for debugging.

Android Debugging Bridge (ADB)

Android Debugging Bridge or ADB is one of the most versatile components of the Android SDK. Although the name suggests that it is a debugging tool, it is actually an all-in-one tool which lets you manage the state of an Android device or emulator. ADB is a client-server program which includes a client (invoked by the command adb), a server which runs as a background process on the development machine, and a daemon which runs as a background process on each emulator or device instance. ADB is located in the <sdk>/platform-tools directory.

On retail devices, you will need to enable Settings>Applications>USB debugging to enable ADB on the device side.



Enabling USB debugging

Common tasks

1. Querying for emulator/device instances

Before you do anything with ADB, it is always a good idea to check if the device is being detected by it. You can use the 'devices' command to get a list of connected Android devices.

```
$ adb devices
List of devices attached
HT0CHRX08211    device
emulator-5554  device
```

In this example it shows that there are two Android devices connected to the ADB server: one device and an emulator. ADB also prints the serial number of the device, which is not be confused with the device's actual serial number. In this case the serial number is a string created by ADB to uniquely identify an emulator/device instance by its console port number. You can use the serial number to direct an adb command to a particular device using the option '-s':

```
adb -s <serialNumber> <command>
```

2. Installing applications

You can install Android applications (APK files) using the install command. For example :

```
$ adb install mypackage.apk
```

3. Port forwarding

Port forwarding can be achieved using the 'forward' command:

```
$ adb forward tcp 6100 tcp 7100
```

In this example we forward the host port 6100 to emulator/device port 7100. It comes in very handy when debugging network applications.

4. Copying files to and from an Android device

You can use the 'pull' command to copy a file or directory from a device:

```
$ adb pull <remote file/folder>
<local path>
```

You can use the 'push' command to copy a file or directory to a device:

```
$ adb push <local file/folder>
<remote path>
```

6. Accessing device shell environment

You can use the 'shell' command to access the Android device shell environment.

```
$ adb shell
```

7. Starting debug tools

You can use ADB to start useful debug tools such as logcat, bugreport and JavaTM Debug Wire Protocol (JDWP).

For example, the following command will start logcat:

```
$ adb logcat
```

Android logging system (logcat)

A logging system is crucial for application debugging. Android provides a sophisticated and centralised logging system which can be accessed by the logcat command. Logcat dumps a log of system messages such as stack traces (in the event of application crash) and application messages.

You can start logcat by command:

```
$ adb logcat
--------- beginning of /dev/log/sysem
D/StatusBarPolicy( 1461): cable
plugged, mPluggedBatteryLevel=88
V/NotificationService( 1385):
Charging...
D/WifiService( 1385): [smart wifi]
ACTION_BATTERY_CHANGED pluggedType: 2
I/ActivityManager( 1385): Starting
 activity: Intent { flg=0x10000000
cmp=com.htc.android.psclient/
.UsbConnectionSettings } from pid 1543
W/Vold    ( 1227): Ignoring unknown
switch 'msm72k_udc'
D/Vold    ( 1227): USB connected
D/Vold    ( 1227): Share method ums
now available
I/StorageNotification( 1461): UMS
connection changed to true (media
state mounted)
D/StatusBarService( 1461):
addNotification notification = Sta
```

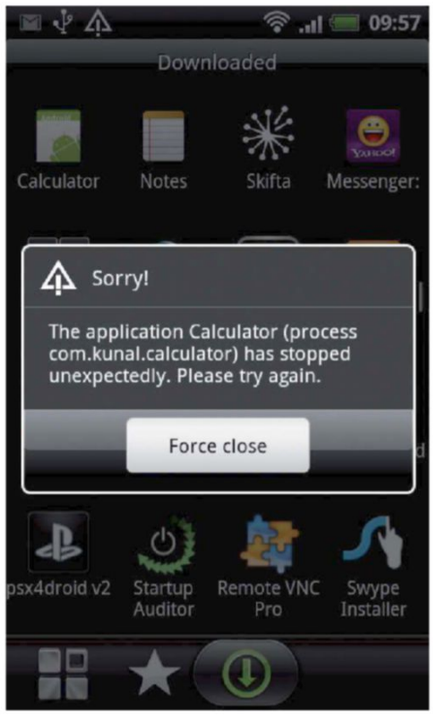


Fig 1 A typical Android application crash

```
tusBarNotification(package=android
 id=17040240 tag=null notification
=Notification(vibrate=null,sound=
null,defaults=0x0,flags=0x2)), key =
android.os.BinderProxy@405fece0,
 fullScreenIntent = null, contentIntent
= PendingIntent{405fede0: android.
os.BinderProxy@4071d720}
```

When an application crashes on the device, you do not see much information there (**Fig 1**).

However, if you look closely, you can find the exact details in the logcat log

```
V/Performance(12830): <<AppBin
dData{appInfo=ApplicationInfo
{4051a4f0 com.kunal.calculator}}
handleBindApplication, 9ms
D/AndroidRuntime(12830): Shutting
down VM
W/dalvikvm(12830): threadid=1:
thread exiting with uncaught
exception (group=0x4001d5a0)
E/AndroidRuntime(12830): FATAL
EXCEPTION: main
E/AndroidRuntime(12830): java.
lang.RuntimeException: Unable
 to instantiate activity
ComponentInfo{com.kunal.calculator/
com.kunal.calculator.MainActivity}:
java.lang.ClassNotFoundException:
com.kunal.calculator.MainActivity in
loader dalvik.system
.PathClassLoader[/data/app/com
.kunal.calculator-1.apk]
E/AndroidRuntime(12830):        at
android.app.ActivityThread.perf
ormLaunchActivity(ActivityThread.
java:1738)
E/AndroidRuntime(12830):
at android.app.ActivityThread
.handleLaunchActivity
..........
..........
E/AndroidRuntime(12830): Caused by:
java.lang.ClassNotFoundException:
com.kunal.calculator.MainActivity
 in loader dalvik.system
.PathClassLoader[/data/app/com
.kunal.calculator-1.apk]
E/AndroidRuntime(12830):
at dalvik.system.PathClassLoader
.findClass(PathClassLoader.java:240)
E/AndroidRuntime(12830):        at
java.lang.ClassLoader.
loadClass(ClassLoader.java:551)
E/AndroidRuntime(12830):
at java.lang.ClassLoader.
loadClass(ClassLoader.java:511)
E/AndroidRuntime(12830):        at
android.app.Instrumentation.newActiv
ity(Instrumentation.java:1061)
E/AndroidRuntime(12830):        at
android.app.ActivityThread.perfo
rmLaunchActivity(ActivityThread
.java:1730)
E/AndroidRuntime(12830):        ...
11 more
W/ActivityManager( 1385):   Force
finishing activity com.kunal
.calculator/.MainActivity
D/Rosie   ( 1507): Activity.
onWindowFocusChangedfalse
```

If you look at the stack trace closely, you will notice the following

```
E/AndroidRuntime(12830): java.
lang.RuntimeException: Unable
to instantiate activity
ComponentInfo{com.kunal.calculator/
com.kunal.calculator.MainActivity}:
java.lang.ClassNotFoundException:
com.kunal.calculator.MainActivity in
loader
```

It seems Android Runtime was not able to locate the class com.kunal.calculator.MainActivity.

Logcat message types

In the stack trace you will notice E, W, D, V etc. These letters indicate the message type thrown by logcat. The following are the types of log messages:

[E]rror: An unexpected critical or non-recoverable failure happened.

[W]arning: Something bad happened, but it was handled gracefully.

[I]nfo: An important event occurred.

[D]ebug: Something happened that may be useful in isolating a problem.

[V]erbose: Something occurred in the normal course of operation that was expected.

Writing to logcat

You can also use logcat for your own application. In fact, it is recommended. You can use an Android utility API called Log to do so. Log is a logging class that you can utilise in your code to print out messages to the logcat. Common logging methods include:

```
v(String, String) (verbose)
d(String, String) (debug)
i(String, String) (information)
w(String, String) (warning)
e(String, String) (error)
```

For example:

@code snippet

```
import android.util.Log;
 protected void
onListItemClick(ListView l, View v,
int position, long id) {
        Intent i = new
Intent(Intent.ACTION_SEND);
        i.setType("text/plain");
        i.putExtra(Intent.EXTRA_
TEXT, items[position]);
        Log.i("ListActivity",
"onListItemClick() - sending text "
+ items[position]);
        startActivity(i);
    }
```

Logcat output

```
I/ListActivity(14013):
onListItemClick() - sending text
Let's go for beers.
```

Advisor

Kunal Deo is a veteran open source developer. Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Fully working code for all the examples is available online at **www.linuxuser.co.uk**. Simply search for 'Debugging your Android applications'.

Code profiling with Traceview

The Android platform creates a do or die world for applications. If your app is not responding to the user at a certain point, an ANR (Application Not Responding) dialog is presented to the user. Users can use this dialog to kill the app or wait for it to respond. Either way, it is a bad thing. ANR is caused in the following events:

1. No response to an input event (eg key press, screen touch) within 5 seconds.

2. A BroadcastReceiver hasn't finished executing within 10 seconds.

An application not responding to a user does not necessarily mean a bug in the application. For example, if your application spends too much time building an elaborate in-memory structure, or perhaps computing the next move in a game, the system will conclude that your application has hung. So you see, even though it is not a coding or logic error, it is as good as having a bug in your code. And fixing it is very important. If you know what is causing the problem in these cases, the recommended approach is to create a child thread and do most of your work there. This keeps the main thread (which drives the user interface event loop) running and prevents the system from concluding that your code has frozen.

In a traditional Android application, finding out what is causing the problem can be a difficult task in itself. That's where code profiling comes into the picture.

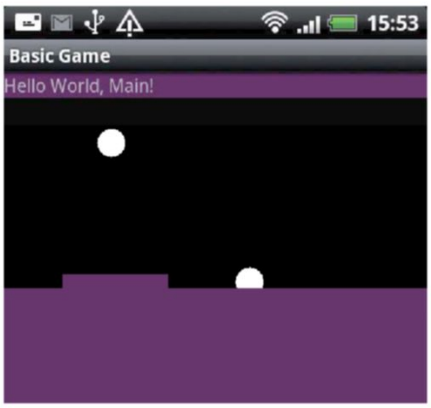


Fig 2 Slow game

Hands on

In this section we will try to isolate a flaw in our Android game which is causing a problem. The application we are using here is an icebreaker-type game which is running very slow (**Fig 2**). Even on our test device, an HTC Desire HD, the application runs very slowly.

Warning: This code sample is best practised on a real device with an SD card installed in it for performance reasons. If you are using an Android emulator, make sure you have created it with the SD card support.

Let's investigate now:

1. Running the game: Open the project and run it in the device or emulator. The project is called 'Basic Game' and can be found at **www.linuxuser.co.uk** (simply search for 'Debugging your Android applications'). Try playing the game. You will see that the game has very poor performance and a very low frame rate.

2. Enable profiling/tracing: We need to isolate and fix this problem. The first step will be enable the tracing in the code. Android SDK provides the class android.os.Debug, which can be used to enable tracing in the app. There are two main methods to keep in mind: to start, use 'startMethodTracing("base name")'; to stop tracing, use 'stopMethodTracing()'. The startMethodTracing command creates a file with a basename.trace file in the /sdcard/ folder, which can then used with the traceview command for investigation. You can put these methods anywhere in your code to start and stop tracing accordingly. In this case we will be calling startMethodTracing() in the activity's onCreate() method, and call stopMethodTracing() in that activity's onDestroy() method.

Edit Main.java as follows:

```
package com.example.basicgame;
import android.app.Activity;
import android.os.Bundle;
import android.os.Debug;
public class Main extends Activity {
@Override
    public void onCreate(Bundle
savedInstanceState) {
   Debug.startMethodTracing("lud");
   super.onCreate(savedInstanceState
);
        setContentView(R.layout.
main);
    }
protected void onDestroy() {
        super.onDestroy();
        Debug.stopMethodTracing();
    }
```

```
}
```

3. Running the application: If you try running the code now, the app will crash instantly. When you look at the logcat output, you may notice the following:

```
E/AndroidRuntime(15515): FATAL
EXCEPTION: main
E/AndroidRuntime(15515): java.lang.
RuntimeException: Unable to start
activity ComponentInfo{com.example
basicgame/com.example.basicgame
.Main}: java.lang.RuntimeException:
 Unable to open trace file '/sdcard
/lud.trace': Permission denied
```

This means that you do not have the permission to write to the SD card. In the next step we will fix this.

4. Setting up application permissions: To enable write permissions to the SD card, open AndroidManifest.xml and insert the highlighted code.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://
schemas.android.com/apk/res/android"
      package="com.example.basicgame"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@
drawable/icon" android:label="@string/
app_name">
        <activity android:name=".Main"
                  android:label="@
string/app_name">
            <intent-filter>
                <action android:
name="android.intent.action.MAIN" />
                <category android:
name="android.intent.category
.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:
minSdkVersion="10" />
    <uses-permission android:
name="android.permission.WRITE
_EXTERNAL_STORAGE"></uses-permission>
</manifest>
```

5. Creating and obtaining the trace file: Now we have everything in place, let's start the application to begin tracing. Due to the overhead of tracing, the application will be even slower now. Play for a while (with patience), then press the Back button to exit the app.

Use the ADB 'pull' command to obtain the trace file:

```
$ adb pull /sdcard/lud.trace $HOME/
Desktop/
```

6. Analysing the trace file: To analyse the trace file, we will use the tool called Traceview. Open the trace file with Traceview.

```
$ traceview $HOME/Desktop/lud.trace
```

Top section of the image (**Fig 3**) shows a close-up of the timeline panel. Each thread has its own execution row. Each method is shown in another colour. The thin lines underneath the first row show the extent (entry to exit) of all the calls to the selected method. You can see that Thread-10 dominates the graph.

The bottom section of the image shows a summary of all the time spent in a method. The table shows both the inclusive and exclusive times (as well as the percentage of the total time). Exclusive time is the time spent in the method. Inclusive time is the time spent in the method plus the time spent in any called functions.

You will notice that the Ball.draw method is consuming 99.6% of exclusive time spent. It means that there is something wrong with our Ball.draw method.

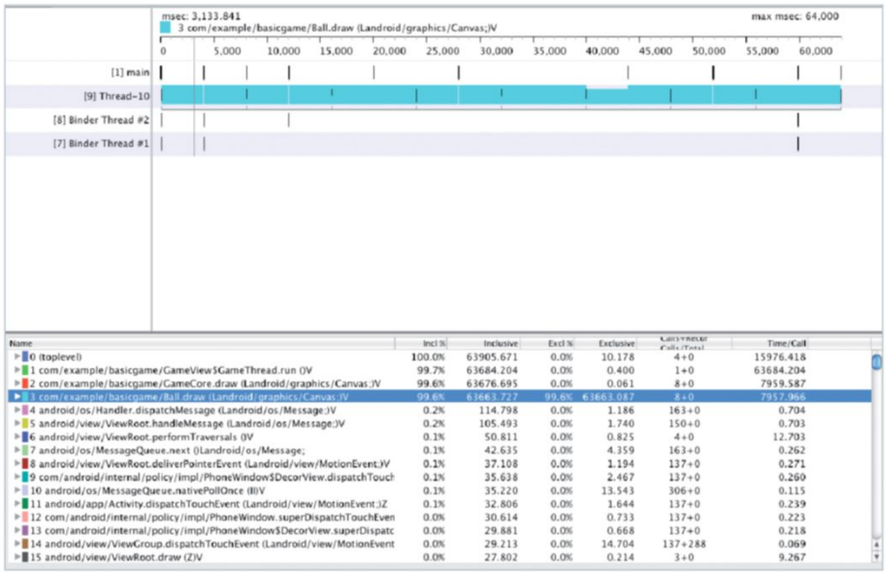


Fig 3 Traceview in action

> **"Debugging isn't always about fixing logical errors"**

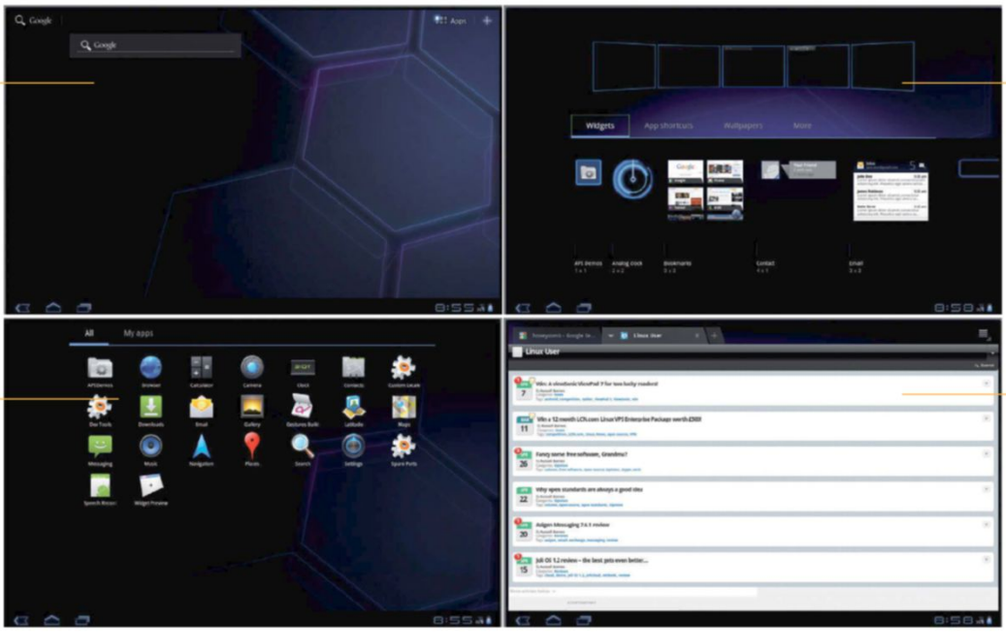
7. Fixing the problem: Let's open the Ball.draw method...

```
void draw(Canvas canvas) {

      int count = 1;
       while (count < 11111111) {

            count++;
       }
       // draw the ball circle on
the screen
       canvas.drawCircle(position.
x, position.y, radius, paint);
}
```

There is an unnecessary 'while' loop that runs for no reason just before the actual canvas.drawCircle method. Remove the while loop and remove the tracing code. Save the project and run the application again. The game should give you a smooth experience now.

Real-world problems would be more complex than the ones presented in the sample code, but tracing/profiling code will help you pinpoint the areas of the problems.

So that was an advanced intro into Android application debugging. As you might have learned, it is not always about fixing logical errors. Sometimes it is also about optimising performance. We have not covered an excellent tool called Dalvik Debug Monitor Server (DDMS) due to space constraints, but we will come back to it in later issues.



Convert your smartphone Android applications to Honeycomb tablet apps

Honeycomb (Android 3.0) is set to become the most popular open source tablet platform. Read on to secure your piece of the pie

Tablets are here. Not that they weren't here before, but this time they are really useful, attractive and pack lot of punch for the price. Tablets are replacing netbooks, providing a premium experience in the same price range. But it is not the price that is actually driving the tablet success: it is the software that is making the difference. Back in the day, Microsoft released 'Microsoft Windows for Pen Computing', while in recent times it tried to sell the same Windows operating system with a pen input but never succeeded. Nowadays it is a different story. There are tablet operating systems written from scratch, with the touch interface being at the core of the user experience. Apple's iOS was the first to pioneer this with the iPad. Now Google is trying to do the same with is first Android operating system designed for tablets, called Honeycomb.

Although in its nascent stage, Honeycomb is growing at an exponential rate. More and more Honeycomb tablets are being released from the likes of Motorola, Samsung and LG. For developers it is an excellent opportunity, a new market to go after. Existing apps can be converted for tablet devices with a little effort. Also, most of the tablet applications are priced higher than smartphone counterparts.

Advisor

Kunal Deo is a veteran open source developer.  Currently he is leading two open source projects: WinOpen64 and KUN Wiki. He is also a KDE developer. He has contributed to many open source projects, including KDE-Solaris, Belenix and Openmoko

Resources

Android SDK Manager

http://www.developer.android.com/sdk/index.html

01 Screen size and resolution

Naturally, tablets provide more screen estate than smartphones. The typical resolution for most of the Honeycomb tablets is 1280x720. You will have to rethink your application to suit a larger screen. Sometimes it is just a matter of providing higher-resolution resources for your application. Other times it requires more complex modifications. The Honeycomb SDK itself will provide a lot of help in this regard, which we will discuss later in this feature.

02 Using API Level 11

Honeycomb SDK is defined under API Level 11. Setting up your application's minimum system version to 11 will set your application build against the Honeycomb SDK. It will also enable the new holographic theme for each of your activities. The holographic theme is the standard theme for Honeycomb. It includes new designs for UI elements, system widgets and overall appearance in general.

You can set the API Level by setting your application's manifest to android: minSdkVersion="11".

Example :

```
<manifest ... >
    <uses-sdk android:
minSdkVersion="11" />
    <application ... >
        ...
    <application>
</manifest>
```

03 Resources

Nearly all Android applications use images. These images will be chosen from the /res/drawables or /res/drawables-mdpi directory. If you do not resize these images for the new higher resolution, they will appear only in their original size. You could also opt for image scaling. The most preferable scale type in this scenario is centerCrop – this will scale the image uniformly while maintaining the aspect ratio.

Example:

```
<ImageView
   android:src="@drawable/
background"
   android:layout_width="match_
parent"
   android:layout_height="match_
parent"
   android:scaleType="centerCrop"
/>
```

If your image has repeating patterns, you can also opt for tiling.

Example:

```
<LinearLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_
parent"
    android:layout_height="fill_
parent"
    android:background="@drawable/
background"
    >
```

Now create the background.xml file in your res/drawable folder:

```
<bitmap
    xmlns:android="http://schemas.
android.com/apk/res/android"
    android:src="@drawable/
baseimage"
    android:tileMode="repeat"
    android:filter="true"
    android:dither="true" />
```

04 Text size

High resolution and a bigger physical screen size may cause your text to get lost or appear small. An easy way to fix this is to increase the font size. If you want to maintain the backward compatibility with smaller screens, create two different <dimen> tags and define a font size with two different values – one as a default text size and one for the text size on an extra-large screen. These tags should be placed inside the dimens.xml file under /res/values and /res/values-xlarge directory.

Example of dimen tag :

```
File Path,  /res/values/dimens.xml

<resource>
   <dimen name="font_size">16sp</
dimen>
</resource>

File Path, /res/values-xlarge/
dimens.xml

<resource>
   <dimen name="font_size">23sp</
dimen>
</resource>
```

Now you can use the variable @dimen/font_size in your application:

```
<style name='titleText" parent="@
android:style/TextAppearance">
    <item name="android:textSize">@
dimen/font_size</item>
    <item name="android:
textColor">#000000</item>
</style>
```

05 Making your application available only to Tablets

At the time of writing, there is no separate section for tablet applications in the Android Market. However, you can limit the availability of your application on tablets using the <supports-screens> manifest element. Android Market will read this manifest element and use it to make sure that the application is only available to the mentioned device type.

For example, the following configuration makes your application available to only extra-large screens, ie tablets.

```
<manifest ... >
    ...
    <supports-screens android:
smallScreens="false"
                      android:
normalScreens="false"
                      android:
largeScreens="false"
                      android:
xlargeScreens="true" />
    <application ... >
        ...
    <application>
</manifest>
```

06 OpenGL hardware acceleration

Most of the Honeycomb tablets on sale (or announced) boast a powerful GPU (graphical processing unit). A very prominent example of this is the GeForce GPU that is being shipped with Nvidia Tegra 2-based tablets such as the Motorola Xoom, Samsung Galaxy Tab 10.1 and Asus Eee Pad Slider. Honeycomb provides a hardware-accelerated OpenGL renderer that utilises the GPU to provide better performance on most 2D graphics operations.

To enable hardware-accelerated rendering in your application, set android:hardware Accelerated="true" in your manifest's <application> element or for individual <activity> elements. While most of the Android tablets should have a powerful GPU, not all will have the same power. You run your application on

Using Adobe AIR to develop Honeycomb applications

Adobe AIR is a cross-OS runtime that lets developers combine HTML, JavaScript, Adobe Flash and Flex, and ActionScript to deploy rich internet applications (RIAs) on a broad range of devices including desktop computers, netbooks, tablets, smartphones and TVs. If Java is giving you headaches, you can use Adobe AIR to develop Honeycomb apps. Adobe AIR on Honeycomb supports GPU-based acceleration and tightly integrates with the Android Browsing engine. Apps written for AIR on one platform can be run unmodified (in most cases) on other supported platforms as well. Creating apps using AIR is very easy.

1.Create application descriptor file (HelloWorld-app.xml)

```
<?xml version="1.0" encoding="UTF-
8"?>
<application xmlns="http://
ns.adobe.com/air/application/2.6">
    <id>samples.android.
HelloWorld</id>
    <versionNumber>0.0.1</
versionNumber>
    <filename>HelloWorld</
filename>
    <initialWindow>
        <content>HelloWorld.swf</
content>
    </initialWindow>
    <supportedProfiles>mobileDevic
e</supportedProfiles>
</application>
```

2. Create application code, which shows text Hello World (HelloWorld.as)

```
package
{
    import flash.display.Sprite;
    import flash.text.TextField;

    public class HelloWorld
extends Sprite
    {
        public function
HelloWorld()
        {
            var textField:
TextField = new TextField();
            textField.text =
"Hello, World!";
            stage.addChild(
textField );
        }
    }
}
```

3. Compile the application

```
$ amxmlc HelloWorld.as
```

4. Create APK package file

```
$ adt certificate -validityPeriod
25 -cn SelfSigned 1024-RSA
sampleCert.pfxsamplePassword

$ adt -package -target apk -
storetype pkcs12 -keystore
 sampleCert.p12 HelloWorld.apk
 HelloWorld-app.xml HelloWorld.swf
```

5. You can deploy this app on a Honeycomb device with AIR runtime installed.

the relevant hardware devices to make sure that hardware acceleration is working fine.

07 Landscape orientation

Most of the tablets by default use landscape orientation as the default. If your original app only supported portrait orientation, you should update it for landscape orientation as well. You can either use a layout that works in both portrait and landscape orientation, or you can provide alternative layout resources for each orientation.

You can use the resource directories as follows:

```
res/layout/             -- default
res/layout-port/        -- portrait
for any screen size
res/layout-xlarge/      -- any
orientation on xlarge screens
res/layout-xlarge-land/ -- landscape
on xlarge screens
```

You should also handle all orientation changes. In an event where the user rotates the screen, the system restarts the current activity by calling onDestroy() and onCreate() in immediate succession. You can handle the orientation changes in the same events.

08 Using of variable features

Because of the open nature of Android, not all devices are created the same in terms of hardware configuration. One common example is that there are 3G and Wi-Fi tablets on the market. 3G tablets incorporate 3G (or telephony) hardware, but Wi-Fi-only tablets do not have the telephony hardware to save on cost. Other hardware features that may be optional are Bluetooth, gyroscope, GPS, barometer etc. You can define a hardware feature using the element <uses-feature>. You can use <uses-feature> to define both mandatory and optional features. If a hardware feature is defined as mandatory, the application will not be installed on any hardware which lacks this feature.

Example:
Mandatory hardware feature...

```
<uses-feature android:name="android.
hardware.telephony" />
```

An optional hardware feature can be defined by adding android:required="false".

```
<uses-feature android:name="android.
hardware.telephony" android:
required="false" />
```

You can also use PackageManager. hasSystemFeature() to determine if a particular hardware feature is available or not.

```
PackageManager pm =
getPackageManager();
boolean hasTelephony = pm.hasSyst
emFeature(PackageManager.FEATURE
_TELEPHONY);
```

Fig 1 Action Bar In the Email application, it provides action items for composing new emails and refreshing the inbox

09 Action Bar

The Action Bar (**Fig 1**) is a widget for activities that replaces the traditional title bar at the top of the screen. Using the Action Bar properly is crucial for any good tablet application. You can use the Action Bar for a lot of cool things such as displaying the items from the options menu, providing tabs for navigating between fragments, and drop-down lists for navigation. You can also use the Action Bar to provide interactive 'action views' in place of action items.

The Action Bar is available to all Honeycomb applications. Adding an Action Bar is as simple as targeting your application to API Level 11.

For example :

```
<manifest xmlns:android="http://
schemas.android.com/apk/res/android"
      package="com.example.
helloworld"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:
minSdkVersion="4"
              android:
targetSdkVersion="11" />
    <application ... >
        ...
    </application>
</manifest>
```

Example: The following code declares a menu item as an action item in a menu resource file.

```
<?xml version="1.0" encoding="utf-
8"?>
<menu xmlns:android="http://schemas.
android.com/apk/res/android">
    <item android:id="@+id/menu_add"
          android:icon="@drawable/
ic_menu_save"
          android:title="@string/
menu_save"
          android:showAsAction="ifRo
om|withText" />
</menu>
```

10 Optimising the user interface using fragments

Since tablets have a greater screen size, this provides room for more user interface elements, unlike smartphones where you can only interact with one screen at a time. Fragments provide a way to use these small activities to form a single activity (**Fig 2**).

You can create fragments using the Fragment class. The Fragment class is part of the Honeycomb SDK.

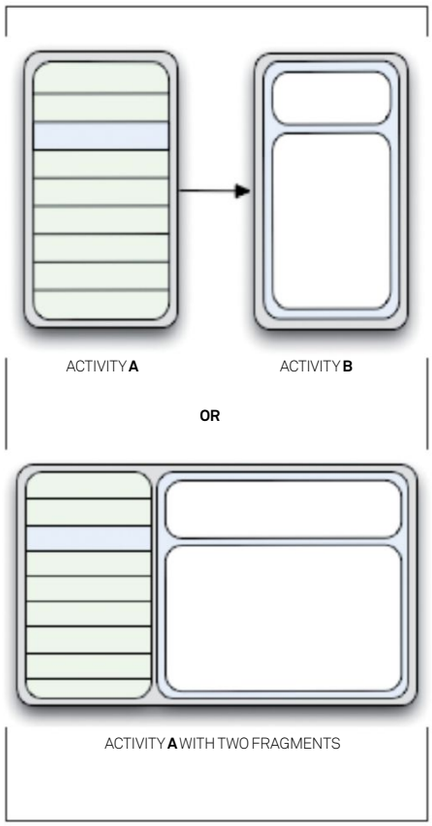
Example: Part of MainAcitvity.Java

```
final FragmentManager fm =
getFragmentManager();
final TitlesFragment f =
(TitlesFragment) fm
.findFragmentById(R.id.frag_title);
final View titlesView = f.getView();
...
 if (shouldShow) {
          fm.beginTransaction().
show(f).commit();
...
 fm.beginTransaction().hide(f).
commit();
```

Activity layout example :

```
<FrameLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    android:layout_width="match_
parent" android:layout_
height="match_parent">
    <fragment class="com.example.
android.apis.app.FragmentLayout$Titl
esFragment"
            android:id="@+id/titles"
            android:layout_
width="match_parent" android:layout_
height="match_parent" />
</FrameLayout>
```

Fig 2 Using fragments How two UI modules are combined into a single activity



> **Using the Action Bar properly is crucial for any good tablet application. You can use it for a lot of cool things such as displaying the items from the options menu, or drop-down lists**

In this tutorial we have covered just a fraction of the new features available in the Honeycomb SDK, but we think that this should be sufficient to get you started with the SDK.

Since the market is new for Android tablet applications, this is an awesome time to be a developer of them since this will give you early visibility in the Android tablet apps marketplace.

Figure out what is happening to your code with OProfile

Even the best programmer needs help sometimes. OProfile can offer that help by letting you know how your program is using resources from the kernel level, right up to program level

```
File  Edit  View  Search  Terminal  Help
CPU: AMD64 processors, speed 3015.51 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Cycles outside of halt state) with a unit mask
of 0x00 (No unit mask) count 100000
CPU_CLK_UNHALT...|
  samples|      %|
------------------
    72959 97.0755 /no-vmlinux
     1024  1.3625 /lib64/libc-2.5.so
      451  0.6001 /bin/bash
      254  0.3380 /usr/bin/oprofiled
      108  0.1437 /lib64/ld-2.5.so
       96  0.1277 /usr/bin/opreport
       72  0.0958 /opt/n1ge/bin/lx24-amd64/sge_execd
       41  0.0546 /lib64/libpthread-2.5.so
       31  0.0412 /usr/lib64/libstdc++.so.6.0.8
       22  0.0293 /bin/gawk
       20  0.0266 /lib64/libcrypto.so.0.9.8b
       14  0.0186 /usr/sbin/ntpd
       13  0.0173 /usr/sbin/sshd
       11  0.0146 /bin/grep
        8  0.0106 /usr/sbin/irqbalance
        7  0.0093 /sbin/init
        6  0.0080 /lib64/libglib-2.0.so.0.1200.3
:
```

A report of all of the running programs on your system

Contrary to how most programmers rate their own code, no piece of software is perfect, especially straight out of the gates. This is where the fine art of debugging comes into play during the development cycle. Once you have your code working correctly, you can then start looking at making your code blazingly fast. This is where the equally fine art of profiling comes in. There are many tools available to profile your code and generate statistics on where your code needs work, such as gprof. These kinds of tools run in user space, and so have inherent limitations on what they can record. If you want to profile your code at a very deep level, including profiling libraries and hardware effects, like cache misses, then you should look at OProfile.

OProfile is a profiling tool that runs at the kernel level. Because of this, you can profile anything at all on the system, including the kernel itself. If you do want to profile the kernel, you will need to hold onto the vmlinux file when you compile your kernel. You can't use the compressed version, vmlinuz, which is what you actually boot up off of. You don't normally have this file when you install most distributions, so you may have to install an extra package, or compile your own kernel. This article won't deal with the kernel. We're going to look at using OProfile to get profiling information on our own code in order to squeeze the most out of it. To do this, we will need to use the command-line option '--no-vmlinux'.

In terms of system requirements, you need to be running at least a 2.2 kernel. If you use 2.2 or 2.4 kernels, you are limited to profiling on 32-bit x86 and IA64 CPUs. Running a 2.6 kernel, you can profile on x86, IA-64, Alpha, MIPS, ARM, x86-64, sparc64, ppc64 and several others. You also need modutils 2.4.6 or later, in order to be able to load the kernel module for OProfile. Since you need to load a kernel module, you will also need to run this as root. We should mention here the concept of sudo. Normally, you can run programs with root permissions by using sudo, but this is not suggested in this case due to an untrusted search path vulnerability. This means that users could end up modifying the PATH environment variable and running malicious executables, so you have been warned.

OProfile launches a daemon (oprofiled) to actually collect the information on what is happening with your code. **To start the collection process, you would execute**

```
opcontrol --no-vmlinux --start
```

By default, OProfile stores the sample files in the directory /var/lib/oprofile. If you want to store these samples in another directory, you can use the command-line option '--session-dir=', but if you do, then all subsequent OProfile commands will also need to include this option so that the sample files can be found. **To store the samples in the subdirectory oprofile in your home directory, you would use**

```
opcontrol --no-vmlinux --session-dir=$HOME/oprofile --start
```

When you finish with your profiling session, you can stop the sampling with '--stop', or completely shut down OProfile with '--shutdown'. By default, the current profiling session is stored in the subdirectory $SESSION_DIR/samples/current. **Once you are done, you will probably want to clean out the sample data by executing...**

Advisor

Joey Bernard has been using Linux since version 2.0 of Slackware. His day job is helping researchers to do science on Red Hat clusters

```
File  Edit  View  Search  Terminal  Help
[root@cl001 ~]# opreport --demangle=smart --symbols /opt/n1ge/bin/lx24-amd64/sge
_execd
CPU: AMD64 processors, speed 3015.51 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Cycles outside of halt state) with a unit mask
of 0x00 (No unit mask) count 100000
samples  %        symbol name
12       16.6667  cl_com_tcp_open_connection_request_handler
5         6.9444  dispatch
4         5.5556  cl_raw_list_unlock
4         5.5556  cl_thread_get_thread_config
3         4.1667  cl_connection_list_destroy_connections_to_close
3         4.1667  init_array
3         4.1667  rmon_condition
2         2.7778  cl_com_host_list_refresh
2         2.7778  cl_log_list_log
2         2.7778  lGetPosInDescr
2         2.7778  sge_add_str2load_report
1         1.3889  cl_com_get_host_list
1         1.3889  cl_com_tcp_write
1         1.3889  cl_com_trigger
1         1.3889  cl_commlib_handle_connection_ack_timeouts
1         1.3889  cl_commlib_receive_message
1         1.3889  cl_commlib_send_sim_message
1         1.3889  cl_commlib_trigger
```

A report of Sun Grid Engine, looking at its symbols

```
File  Edit  View  Search  Terminal  Help
[root@cl001 ~]# opreport -cl /opt/n1ge/bin/lx24-amd64/sge_execd
CPU: AMD64 processors, speed 3015.51 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Cycles outside of halt state) with a unit mask of 0x
00 (No unit mask) count 100000
samples  %        symbol name
-------------------------------------------------------------------------------
12       16.6667  cl_com_tcp_open_connection_request_handler
  12      100.000  cl_com_tcp_open_connection_request_handler [self]
-------------------------------------------------------------------------------
5         6.9444  dispatch
  5       100.000  dispatch [self]
-------------------------------------------------------------------------------
4         5.5556  cl_raw_list_unlock
  4       100.000  cl_raw_list_unlock [self]
-------------------------------------------------------------------------------
4         5.5556  cl_thread_get_thread_config
  4       100.000  cl_thread_get_thread_config [self]
-------------------------------------------------------------------------------
3         4.1667  cl_connection_list_destroy_connections_to_close
  3       100.000  cl_connection_list_destroy_connections_to_close [self]
-------------------------------------------------------------------------------
3         4.1667  init_array
  3       100.000  init_array [self]
-------------------------------------------------------------------------------
```

A call-graph report of Sun Grid Engine

`opcontrol --reset`

This will erase the data stored in the 'current' subdirectory within $SESSION_DIR/samples. **If you want to save this session data for later analysis, you can do so by using**

`opcontrol --save=TEST1`

which will save the current session data into the subdirectory $SESSION_DIR/samples/TEST1.

So, now that you know how to start, stop and save the profiling data, what can you profile? Using the option '--event=', you can specify which events OProfile should be tracking during the session. **Events are specified using a colon-separated string of the form name:count:unitmask:kernel:user**

name – symbolic event name
count – counter reset value
unitmask – unit mask, as given in the events list
kernel – whether to profile kernel code
user – whether to profile user-space code

> **"OProfile takes a number of samples each second"**

The last three values are optional. If you leave them off, then OProfile sets them to unitmask equal to 0, kernel set to yes and user set to yes. In most cases, this is actually what you want. So you will be left setting the event name and the count value. Depending on the CPU architecture that you are running on, there are different default events that OProfile will track. On an Athlon the default event is CPU_CLK_UNHALTED:100000, whereas on a Pentium 4 it is GLOBAL_POWER_EVENTS:100000. If you want to find out what events are able to be profiled on your system, use '--list-events' to get a list. **The output will look like:**

```
oprofile: available events for CPU
type "Pentium M (P6 core)"
See Intel Architecture Developer's
Manual Volume 3, Appendix A and
Intel Architecture Optimization
Reference Manual (730795-001)

CPU_CLK_UNHALTED: (counter: all)
clocks processor is not halted, and
not in a thermal trip (min count:
6000)
DATA_MEM_REFS: (counter: all)
all memory references, cachable and
non (min count: 500)
```

You may notice in the above example that there is a min count in each description. OProfile is a statistical profiler. This means that it takes a number of samples each second, where the min count is the minimum number that you can set for that particular event. If you want to collect more samples, then you need to set the counter value lower for that event. This means, however, that your system will be spending more time responding to the OProfile daemon, so this means that you will need to do some balancing between sample counts and responsiveness. **If you wanted to get the maximum number of samples for both CPU and memory references, you would use**

```
opcontrol --event=CPU_CLK_
UNHALTED:6000 --event=DATA_MEM_
REFS:500
```

Now that you have all of this data collected, how do you see it? OProfile includes the program opreport to print out the results in a human-readable form. **To get a summary, you can run**

```
opreport --long-filenames
```

On our netbook, we get the results:

```
CPU: Pentium M (P6 core), speed 800
MHz (estimated)
Counted CPU_CLK_UNHALTED events
(clocks processor is not halted, and
not in a thermal trip) with a unit
mask of 0x00 (No unit mask) count
6000
Counted DATA_MEM_REFS events (all
memory references, cachable and non)
with a unit mask of 0x00 (No unit
mask) count 500
CPU_CLK_UNHALT...|DATA_MEM_REFS:500|
samples|      %|  samples|      %|
 ---------------------------------
---
10332 96.7235     28885 97.8290 /no-
vmlinux
94  0.8800       116  0.3929 /lib/
tls/i686/cmov/libc-2.9.so
57  0.5336       137  0.4640 /usr/
bin/Xorg
53  0.4962       128  0.4335 /usr/
lib/xorg/modules/drivers/intel_drv.
so
```

So, you can see where the system is spending most of its time and who is making the most memory references. The option '--long-filenames' prints out the full path to each referenced binary file. OProfile can also print out call-graph information with the command-line option '--callgraph'. If you are more interested in even more detailed information, the command-line option '--details' will print out instruction-level details. For binaries that have no symbol information (ie they were not compiled with the debugging flag turned on), opreport will print out VMA values as raw file offsets for the binary.

There is also a tool provided which can output source code or assembly listings annotated with OProfile results. **In order to see annotated assembly listings for a program, say ls, you would simply run**

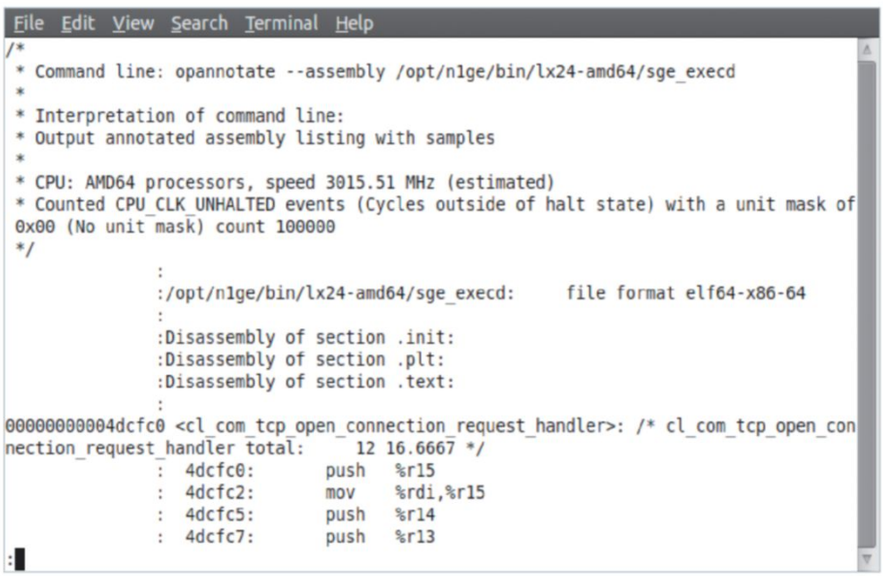
```
opannotate --assembly /bin/ls
```

If you compiled your program with symbol information (ie with the '-g' option), you can get annotated source code by running

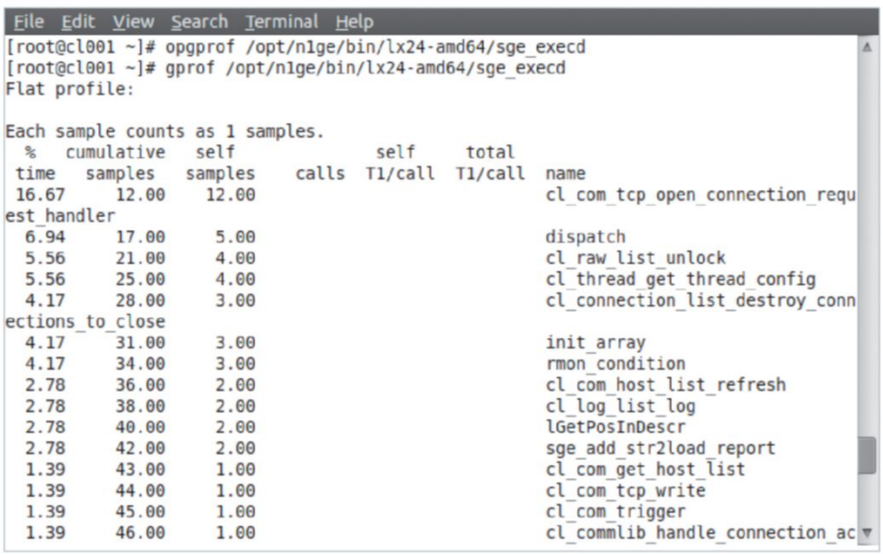
```
opannotate --source ./my_prog
```

The results here may be inaccurate. You need to remember that OProfile is a statistical profiler, so if your program is small, or runs in a very short amount of time, then the likelihood of getting useful results will be low. For larger programs, or for long-running programs, you start to get more and more accurate results.

You may be more familiar with the output produced by gprof. If so, you can get OProfile to output results in this format by using the included tool opgprof. You simply have to hand in the path to the binary image and the file 'gmon.out' will be written out in the current directory. You can then hand this file in to gprof



Annotated assembly code disassembly of a program



Using OProfile statistics in gprof

> **"For larger programs, you start to get more and more accurate results"**

to do your analysis.

The last tool in the OProfile suite is oparchive. This program bundles up all of the executable, debug and sample files into a directory so that you can tar them up and move them to another machine in order to do the actual analysis. **By running**

```
oparchive -o ./archive
```

a full copy of all files necessary to do analysis will be copied into the subdirectory './archive'. You can then move this directory wherever you would like to in order to do further analysis.

Having learned all of this, let's look at a possible scenario. You've been working on some sophisticated number-crunching code and you want to see if the cache is being utilised in the best possible way. **The first thing to do is to look at what cache events can be tracked on your system:**

```
opcontrol --list-events | grep CACHE
```

On our system, we get

```
INSTRUCTION_CACHE_FETCHES: (counter:
all)
INSTRUCTION_CACHE_MISSES: (counter:
all)
DATA_CACHE_ACCESSES: (counter: all)
DATA_CACHE_MISSES: (counter: all)
DATA_CACHE_REFILLS_FROM_L2_OR_
SYSTEM: (counter: all)
DATA_CACHE_REFILLS_FROM_SYSTEM:
(counter: all)
DATA_CACHE_LINES_EVICTED: (counter:
all)
DCACHE_MISS_LOCKED_INSTRUCTIONS:
(counter: all)
DCACHE Misses by Locked Instructions
(min count: 500)
L2_CACHE_MISS: (counter: all)
L2_CACHE_FILL_WRITEBACK: (counter:
all)
INSTRUCTION_CACHE_REFILLS_FROM_L2:
(counter: all)
INSTRUCTION_CACHE_REFILLS_FROM_
SYSTEM: (counter: all)
CACHE_BLOCK_COMMANDS: (counter: all)
```

We're not too interested in the instruction cache, **but more in seeing whether the data cache use is optimised, so we'll look at that:**

```
opcontrol --event:DATA_CACHE_
ACCESSES:500 --event=DATA_CACHE_
MISSES:500 --event=DATA_CACHE_LINES_
EVICTED:500
   opcontrol --no-vmlinux
   opcontrol --start
```

We can now run our code and let OProfile log all of this cache behaviour. **Once our program finishes its work, we can look at the results.**

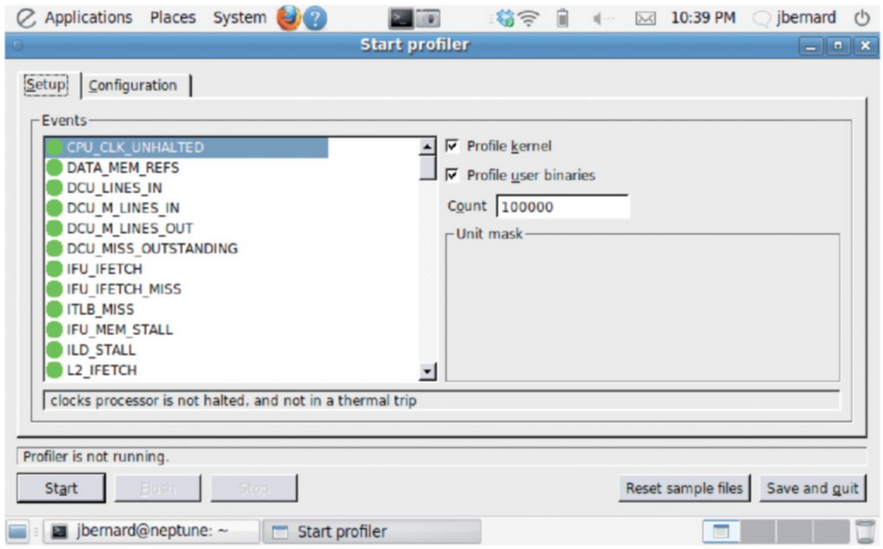


Fig 1 The OProfile start screen, where you can select what to profile

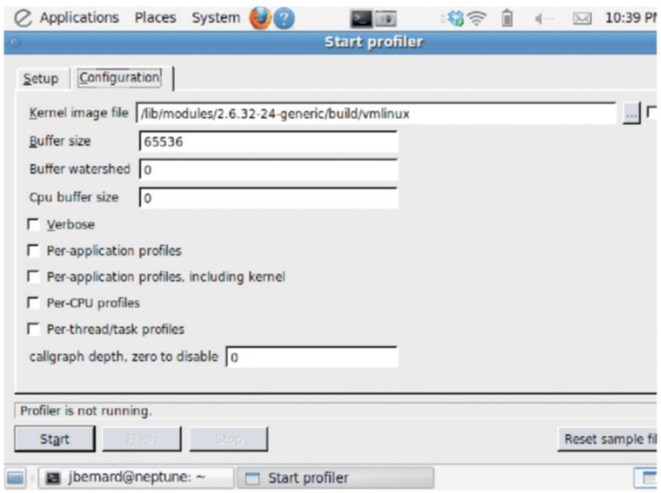


Fig 2 Configuration screen for the OProfile GUI

```
opcontrol --shutdown
opreport -l ./my_prog event:DATA_
CACHE_MISSES
```

This would give us a report of specifically the data cache misses within our program, and where they were happening.

All of the work we've done until now has been at the command line. There is a graphical interface to oprofile available. When it starts up, you can select what events get profiled (**Fig 1**).

The GUI also has a configuration screen where you can set the kernel image file (**Fig 2**), whether you want per-application profiles, whether you want per-CPU profiles, or if you want to set the call-graph depth, among other options.

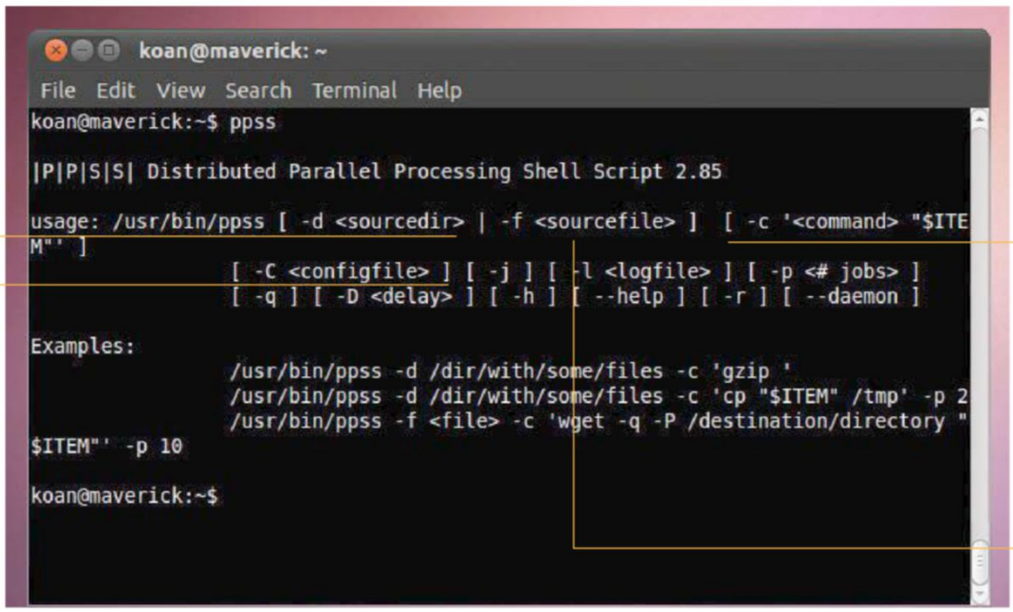
Now you should be ready to go ahead and do some analysing of your own code. But before you do, you should be made aware of a few potential pitfalls. The first thing to be careful of is to not set count levels too low. You may end up leaving your system unresponsive. When you look at annotated source, you may be hit with problems caused by compiler optimisations. The compiler may move code around, causing OProfile to show odd behaviour. You also need to be aware of the hidden cost of instructions. One example is two memory reads, one from L1 and one from memory. Another example would be a mis-predicted branch. Also, OProfile doesn't work very well in a virtual machine (such as VMware). Hopefully this short article will give you one more tool in your arsenal in the quest for the fastest, most efficient code out there.

Tap into the potential of multi-core CPUs

Use Parallel Processing Shell Script to make use of multi-core processors or even divide your workload among computers...

The source directory contains the files which are processed in parallel by PPSS

You can save your special configuration in a configuration file for further use with PPSS



The command to be executed with the file as an argument can be simple or use variables defined by PPSS

Alternatively, you can point to a list of files contained in a file

Resources

PPSS is a Bash script that uses FIFOs, computes MD5 checksums and makes use of regular expressions. Therefore, the requirements are:

bash
mkfifo
md5sum
sed

These programs are installed on most Linux distributions by default. Mac OS X is also supported, and PPSS should work on other UNIX-like operating systems, as long as you use the Bash shell. If you want to make use of the distributed mode of PPSS on multiple hosts, there are some additional requirements:

ssh and sshd
screen

Even on recent multi-core systems, most shell scripts don't use the full power of your computer. Thanks to PPSS (Parallel Processing Shell Script), you can write shell scripts that tap into the potential of current multi-core CPUs. PPSS detects the number of available CPUs and starts a separate job for each core. You can also use PPSS on multiple hosts in a distributed fashion to divide the workload among your computers.

If you want to automate a repetitive task by way of a shell script, chances are you normally use a loop construct. **For example, if you want to convert all your music files in the lossless WAV format to the lossy MP3 format with lame, you'll do something like this:**

```
for f in *.wav; do lame "$f" "${f%.wav}.mp3"; done
```

The problem here is that the lame command handles the WAV files one by one, because the loop construct works sequentially. The result is that only one of your CPU cores is used, even if you have one of those many-core monsters. Just look at the output of top to see this with your own eyes. So you'll begin to wonder: why have you bought such a good CPU when your shell scripts can't make use of them?

> **"PPSS detects available CPUs and starts a separate job for each core"**

Advisor

Koen Vervloesem has been writing about free and open source software, and IT in general, since 2000. He has master's degrees in computer science and philosophy and can be reached at koen@vervloesem.eu

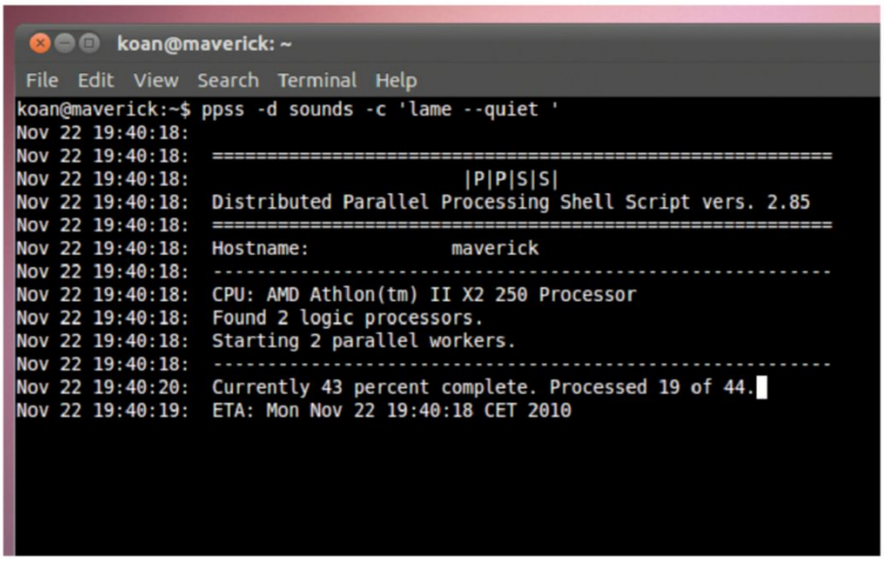


Fig 1 Simple The PPSS equivalent of our loop code

01 Don't do multi-core yourself

Of course, you can program your own parallel loop constructs that can convert multiple music files at the same time, but this is difficult stuff and will take a lot of time to write and debug. Luckily there is already a tool that has done this for you and that lets you focus on your own commands: PPSS (Parallel Processing Shell Script), which you can download on its website: https://code.google.com/p/ppss/.

02 Automatic parallellisation

PPSS is a Bash shell script that executes commands, scripts or programs in parallel. It detects the amount of available CPU cores and starts a separate job on each core. The program even supports Hyper-Threading if it's available on your CPU. In the simplest form, you only have to supply it a source of items (such as a directory with files) and a command that has to be applied to these items.

03 Simple

Our 'for' loop to convert WAV files to MP3 becomes this if we use PPSS instead:

```
ppss -d /path/to/wavfiles -c 'lame '
```

The first thing you'll notice is that this command lacks a loop construct, because PPSS takes care of it, in its smarter way. By default, PPSS handles all files in the directory after the -d option. The -c option contains the command you want to apply to all files (**Fig 1**).

04 Variables

If the command between single quote marks has a trailing space, like in our example, the filename will be appended to the command; for example, -c 'lame ' will execute the command lame on each file in the directory. If the item should not be appended to the command, but inserted somewhere in the middle, you use the variable $ITEM in the command, which will be replaced by the specific file for each job. An example: ppss -d /path/to/files -c 'cp "$ITEM" /destination/dir'.

05 A list of files

If you only want to work on some specific files, you list the file paths in a file and use the -f option with the name of this file. The -f option is, however, more flexible than this: instead of files, each line in the file can be any type of item, eg URLs. **If you want to download a list of files, just fill the file list-of-urls.txt with the URLs and then execute:**

```
ppss -f list-of-urls.txt -c 'wget -q '
```

06 Log files

Now if you execute PPSS, you'll see that it shows how many cores it finds, and it shows a general progress indicator while it does its job. At the end, PPSS tells you the total processing time. During its work, you won't be able to see the output of your command, but you can consult the log file of PPSS itself in ppss_dir/ppss-log-*.txt and any output of the individual jobs in the directory ppss_dir/job_log. These log files show, for each item, how much time it took to process it, whether it was successful and so on (**Fig 2**).

07 Abort and continue

If you interrupt PPSS (eg by pressing Ctrl+C) and run it again with the same options

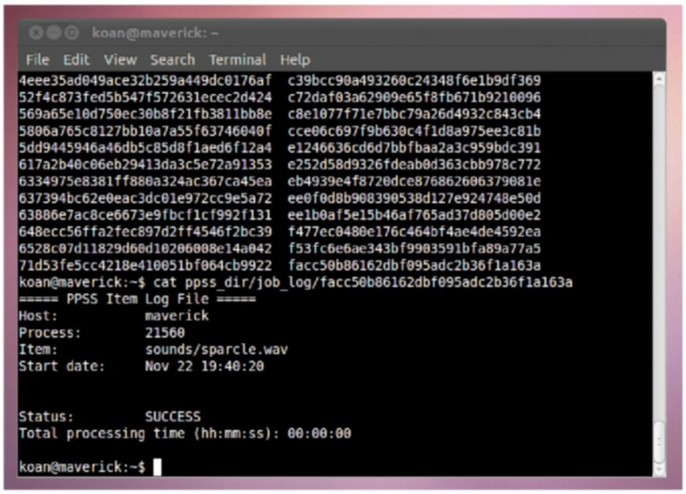


Fig 2 Log files Showing processing time and more

later, it will detect the presence of item log files in the job_log directory and continue where it was left, skipping already processed files. If you want to process all items again, just remove the job_log directory before rerunning PPSS.

08 Not a silver bullet

An import thing to take into account is that PPSS is not a magic solution: it's only useful for jobs that can be easily broken down into separate tasks that can be executed in parallel. Converting a lot of files from one format to another is a prototypical example of this. Other examples are downloading a large number of files and resizing images. In contrast, if different items depend on each other, PPSS is not useful any more.

09 Help

PPSS has a couple of options to tweak its behaviour. If you execute the command without any options, you get some information about the usage, and if you execute it with the -h option, it displays more options with some help. For a full listing of all options, including the distributed mode we'll talk about later in the article, run PPSS with the option --help.

10 Hyper-Threading

Two of these options are interesting if you want to change the number of cores PPSS uses, for whatever reason. With -j you disable Hyper-Threading. For example, an Intel Core i7 quad-core processor supports Hyper-Threading, so by default PPSS starts eight parallel jobs on it. If you disable Hyper-Threading, PPSS only uses four parallel jobs.

11 Fewer jobs

A more flexible option is -p <number>, which allows you to specify exactly how many parallel jobs should be started. You can set this number to less than the number of cores if you don't want PPSS to use all your CPU's resources, eg if it's a desktop system and working on it becomes too sluggish when using all cores for PPSS jobs running in the background.

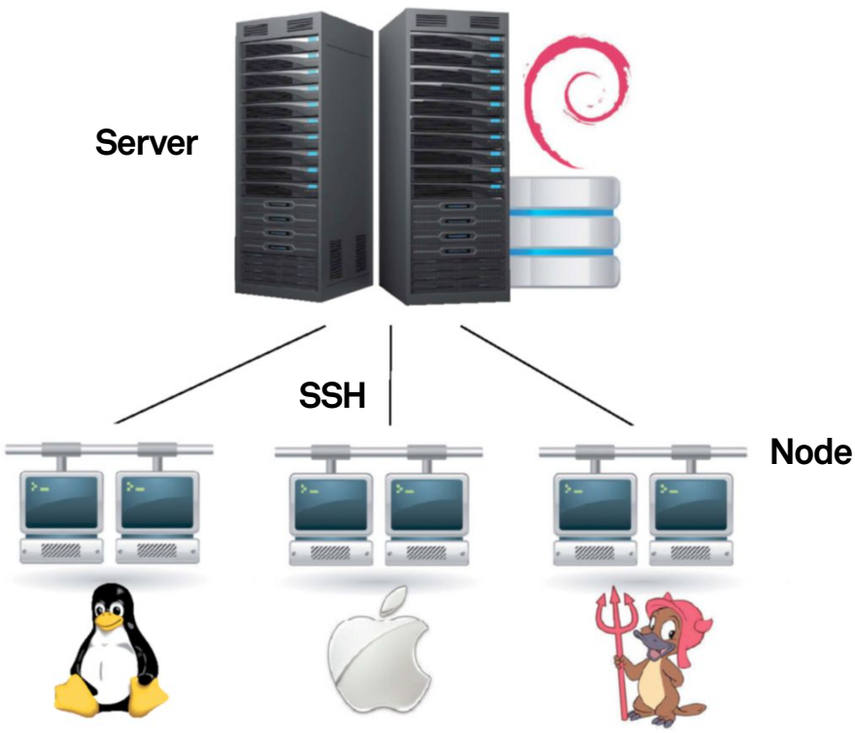


Fig 3 Distributed processing step by step PPSS architecture

12 More jobs

What's maybe less obvious is that the -p option is also interesting for single-core CPUs. For example, if you have to download a bunch of files and process them, you can optimise this in many cases by grouping the download and processing (eg unpacking a tgz file) in one job for each file, and then start a couple of jobs in parallel. Downloading, especially, is an interesting task for PPSS, as this is not bound by the number of available CPUs.

13 Configuration files

Another interesting option is to create a config file with a specific configuration. Just run PPSS with all options you need, but add 'config -C config.cfg' as the first option. This command creates a config file, config.cfg, that contains all other options. Then if you want to run PPSS with these options, you don't have to re-enter all command-line options, but just run 'ppss -C config.cfg'.

14 Distributed processing

PPSS can also be run on multiple computers, processing a single group of items together, like a cluster. All PPSS clients then process a part of the same queue of items, and they communicate with each other through an SSH server which co-ordinates all clients.

15 Distributed processing step by step

To use PPSS in a distributed way, we have to perform the following steps:

1. Set up SSH access on server and nodes.
2. Create a list of all nodes.
3. Create a configuration file for PPSS.
4. (Optional) Create a custom script to be executed.
5. Deploy PPSS to the nodes.
6. Start PPSS on all nodes. (**Fig 3**)

> **"PPSS can also be run on multiple computers, processing a single group of items together, like a cluster"**

16 User 'ppss'
We first create an unprivileged user 'ppss' on the server and on each node: sudo adduser ppss. Then log in as the ppss user on the server and create an SSH key: ssh-keygen. When asked to enter a passphrase, just push Enter for no passphrase. Then create a directory and enter it: mkdir ppss-home && cd ppss-home.

17 SSH keys
Now, add the public SSH key (/home/ppss/.ssh/id_rsa.pub) to the ~/.ssh/authorized_keys file of the 'ppss' user on the server (cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys), and do the same to the nodes:ssh-copy-id -i ~/.ssh/id_rsa.pub node. The latter asks you the password for the ppss user on the node. After this, you should be able to log into each node from the server as the ppss user without password. Then on each node, connect once to the server and manually accept the server signature

18 Nodes
Now create a file on the server with a list of the nodes, **which can be hostnames or IP addresses:**

```
192.168.1.6
192.168.1.7
host.domain.com
...
```

Next create directories for input and output files: mkdir -p /home/ppss/files/input /home/ppss/files/output

19 Distributed configuration
The next step is to create a config file for PPSS. This works the same as setting up a config file for the normal mode, except that the distributed mode requires a lot more options. **An example:**

```
ppss config -C config.cfg -c 'lame -a
"$ITEM" "$OUTPUT_DIR/$OUTPUT_FILE.
mp3" --preset standard --quiet' -d
/home/ppss/files/input -m 192.168.1.8
-u ppss -k ~/ppss-home/id_rsa -n
~/ppss-home/nodes.txt -o /home/ppss/
files/output --upload --download
```

20 File server
So this command creates a config file, config.cfg, and uses the specified lame command on all files in the source directory specified after the -d option. This source directory is located on the server, and these files will be transferred using SCP to the nodes for local processing. The -m option specifies the SSH server that acts as both file server and SSH server for communication between nodes. Because file transfers using SSH can take quite some processing power, you shouldn't use the server as a processing node.

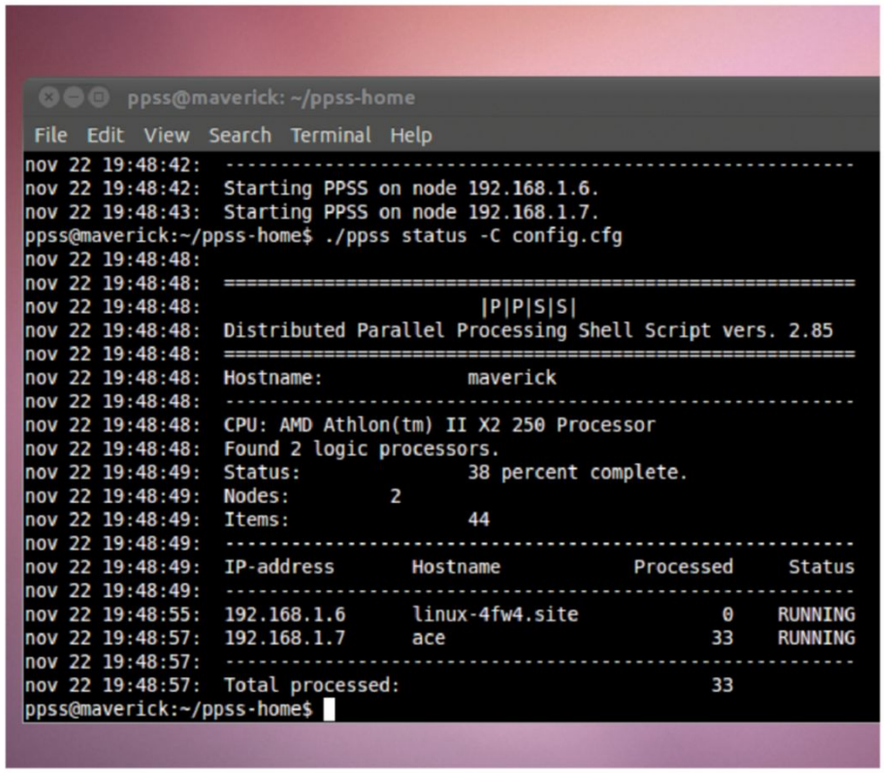


Fig 4 Deployment Deploy PPSS to the nodes to process the files

21 Download and upload
The -u and -k options specify the user and SSH key, respectively. The -n option specifies the file containing all nodes. With --download, the file that is to be processed is copied via SCP from the source directory on the server to a local temporary working directory for local processing. If the --upload option is used, the -o option specifies the destination directory on the server, where the results are uploaded via SCP.

22 Shared files
You can also choose to distribute the files over NFS or SMB: just point the -d option to the mountpoint of the shared directory. The files then look like local files for each node, and then the options --download, --upload and -o can be omitted. Moreover, for complex actions you can specify a script with the -S option instead of a simple command in the -c option. PPSS uploads the script to all nodes.

23 Deployment
Now deploy PPSS to the nodes (**Fig 4**) with 'ppss deploy -C config.cfg', start PPSS on all nodes with './ppss start -C config.cfg', and at any time you can stop, pause or continue PPSS on all nodes with './ppss stop -C config.cfg', './ppss pause -C config.cfg' or './ppss continue -C config.cfg'. You can also check the progress of the distributed computation with './ppss status -C config.cfg'. Note the './', which uses the version of the PPSS script in ppss-home.

24 Result
After this, the resulting files are in your output directory. Hopefully you've seen that PPSS can be very handy if you have to process a bunch of files and want to do this as efficiently as possible. It takes some configuring, but it can save you a lot of time.

Coding with the Qt UI framework

Qt is an excellent UI framework. It is not only the world's most ported UI framework but also the easiest to use...

Qt is an open source cross-platform application and UI framework (or GUI toolkit). Qt is certainly not the only one around – there are many others, but Qt is very different from them. Qt always relies directly on a platform's native APIs to provide native application behaviour, look and feel, and fast performance. Qt is also unique in terms of platform support: it is supported on Linux, Mac OS X, WIndows, Symbian, Maemo, MeeGo, Windows Mobile Android and various other UNIXes.

In this feature we will take a look at Qt's offerings, how Qt works and then we will make a simple application in Qt.

Qt modules

Resources

Qt Development Libraries. You check it using the following command:

```
$ qmake -v
```

version 2.01a
Using Qt version 4.7.0 in /usr/lib

QtCore:
The QtCore module contains the non-GUI but essential classes. QtCore provides some of the most fundamental features of Qt such as loop, signal and slot mechanism. It also includes platform-independent abstractions for Unicode, threads, mapped files, shared memory and regular expressions. All other Qt modules rely on this module.

```
Include directive:
#include <QtCore>
```

QtGui:
QtGui provides many of the GUI-related classes included in the Qt framework. It provides support for various GUI elements such as buttons, text boxes etc. QtGui implements the model-view-controller design pattern.

```
Include directive:
#include <QtGui>
```

QtMultimedia:
QtMultimedia provides classes for low-level multimedia functionality. This includes classes to interact with audio and video devices, support for video surface, frame-rate-related controls etc.

```
Include directive:
#include<QtMultimedia>
```

QtNetwork:
The QtNetwork module contains classes for writing UDP and TCP clients and servers. It includes classes that implement FTP and HTTP clients and support DNS lookups. Network events are integrated with the event loop, making it very easy to develop networked applications.

```
Include directive:
#include <QtNetwork>
```

To link against the module, add this line to your qmake .pro file:

```
 QT += network
```

QtOpenGL:
The QtOpenGL module offers classes that make it easy to use OpenGL in Qt applications. OpenGL is a standard API for rendering 3D graphics. While the standard OpenGL itself provides little or no support for GUI programming elements, Qt provides cross-platform UI elements to develop OpenGL applications.

```
Include directive:
#include <QtOpenGL>
```

To link against the module, add this line to your qmake .pro file:

```
QT += opengl
```

QtOpenVG:
The QtOpenVG module offers classes that support OpenVG drawing. OpenVG is a standard API designed for hardware-accelerated 2D vector graphics. It is aimed primarily at mobile phones, media and gaming consoles.

QtOpenVG support should be configured during Qt configuration. Then you can run Qt apps with OpenVG support:

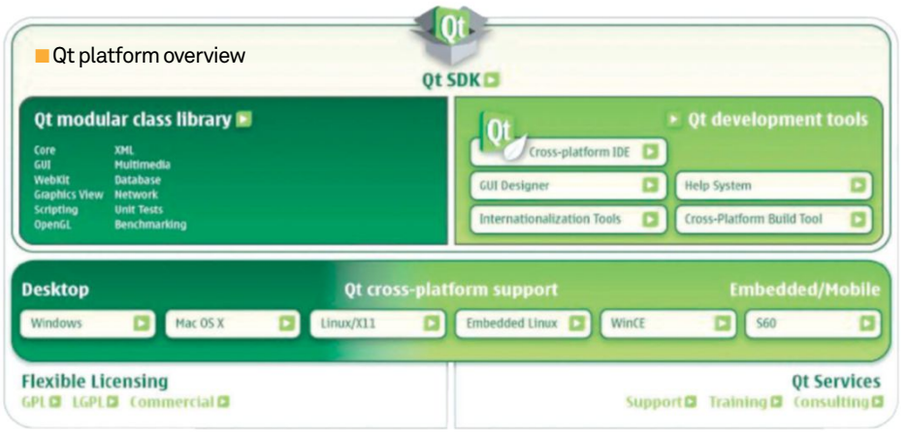
```
$ qtapp -graphicssystem OpenVG
```

QtScript (and QtScriptTools):
QtScript provides support for application scripting with ECMAScript. A very common example of ECMAScript is JavaScript. Other implementations include Objective-J and ActionScript.

```
Include directive:
#include <QtScript>
```

To link against the module, add this line to your qmake .pro file:

```
QT += script
```



QtSql:
QtSql provides classes that help to integrate various open source and commercial SQL databases. QtSql also includes an implementation of the SQLite database.

```
Include directive:
#include <QtSql>
```

To link against the module, add this line to your qmake .pro file:

```
QT += sql
```

QtSvg:
QtSvg includes classes for displaying and creating SVG files. SVG (scalable vector graphics) is an XML-based file format for describing 2D vector graphics.

```
Include directive:
#include <QtSvg>
```

To link against the module, add this line to your qmake .pro file:

```
QT += svg
```

QtWebKit:
The QtWebKit module provides a WebKit-based web browser rendering engine that allows Qt applications to integrate web content. WebKit is an open source rendering engine that is used in Google Chrome, Safari, Android and iOS Mobile Safari browsers.

```
Include directive:
#include <QtWebKit>
```

To link against the module, add this line to your qmake .pro file:

```
QT += webkit
```

QtXml:
The QtXml module implements SAX and DOM interfaces to Qt's XML parser. QtXml also provides a simplified XML parser in the form of a simple XML parser.

```
Include directive:
#include <QtXml>
```

To link against the module, add this line to your qmake .pro file:

```
QT += xml
```

```
Qt3Support:
```

The Qt3Support module provides classes that ease porting from Qt 3 to Qt 4. The Qt team warns that classes in this module are intended to be used in intermediate stages of a porting process and are not intended to be used in production code.

```
Include directive:
#include <Qt3Support>
```

To link against the module, add this line to your qmake .pro file:

```
QT += qt3support
```

Making a Qt app

Now that we have an idea about what all the main Qt modules do, let's use some of them to make a useful but simple app and in the same process learn the nitty gritty of the Qt development process.

The app will be a simple HTTP file downloader. This is not a fully fledged download manager, but can be thought of as a simple GUI version of the Wget download manager. Wget does a lot more, but for the sake of simplicity we will only concentrate on getting the download functionality. This example will also explain the basic process of Qt application development.

Our Hello World App consists of these files:
downloader.pro: This is the Qt Project file for the project.
main.cpp: This file contains the entry point of the application.
httpwindow.cpp: This file describes the main UI window.
httpwindow.h: The header for the application main UI window.
login.ui: The Qt Designer UI file for the login screen (in case the website requires a login).

You can recognise regular C and C++ files here, but there are two special files: downloader .pro and login.ui. Let's look into these file types.
.pro file: This is also known as the qmake project file. qmake is a tool that helps you build Qt software easily on multiple platforms. It generates platform-specific makefiles from rather simple qmake project files (.pro). In addition to generating makefiles, qmake can also generate project files for various integrated development environments (IDEs) such as Xcode (Mac OS X) and Visual Studio (Windows).
.ui file: This defines the user interface elements in the form of XML data. This file can also be created using Qt Designer. Qt Designer is a tool for designing and building graphical user interfaces (GUIs) from Qt components. It provides a visual way to lay out the graphical interface and connect the graphical elements of a Qt application.
Signal and slot mechanism: Signal and slot is a language construct which makes it easy to enable communication between objects without writing too much boilerplate code. The idea is that the controls can send signals containing event information which can be received by other controls or objects. For example, if a user clicks a Close button, we want the window's close() function to be called.

With this in mind let's get back to the code...

HttpWindow class definition

The HttpWindow class displays a window, in which the user can enter the HTTP address of the file to download it. The slots of HtttpWindow are connected to its widgets, and contain the functionality for downloading the file over an HTTP connection. We also connect to signals in QSslError, which detects if the connection requires SSL and throws an appropriate error if Qt is not built with the SSL support.

@code excerpt

```
private slots:
     void downloadFile();
     void cancelDownload();
     void httpFinished();
     void httpReadyRead();
     void updateDataReadProgress(qi
nt64 bytesRead, qint64 totalBytes);
```

```
    void enableDownloadButton();
    void slotAuthenticationRequired
(QNetworkReply*,QAuthenticator *);
 #ifndef QT_NO_OPENSSL
    void sslErrors(QNetworkReply*,c
onst QList<QSslError> &errors);
 #endif
```

HttpWindow class also includes declarations for appropriate UI elements.

@code excerpt

```
 private:
    QLabel *statusLabel;
    QLabel *urlLabel;
    QLineEdit *urlLineEdit;
    QProgressDialog
*progressDialog;
    QPushButton *downloadButton;
    QPushButton *quitButton;
    QDialogButtonBox *buttonBox;
```

We also need to create an instance of QNetworkAccessManager, called qnam. The QNetworkAccessManager class allows the application to send network requests and receive replies.

@code excerpt

```
 QUrl url;
    QNetworkAccessManager qnam;
    QNetworkReply *reply;
    QFile *file;
    int httpGetId;
    bool httpRequestAborted;
```

HttpWindow class implementation

We are importing the 'ui_authenticationdialog.h' file, which is a C++ file generated from the XML .ui file. This file is not part of the project and is generated automatically by the uic tool during the build process. ui_authenticationdialog is called when the URL is password-protected.

@code excerpt

```
 #include <QtGui>
 #include <QtNetwork>
 #include "httpwindow.h"
 #include "ui_authenticationdialog.h"
```

HttpWindow constructor is responsible for setting up the UI. We could have also used the Qt Designer to achieve the same thing. We are also connecting UI elements using the signal and slot mechanism.

@code excerpt

```
buttonBox = new QDialogButtonBox;
buttonBox->addButton(downloadButton,
QDialogButtonBox::ActionRole);
buttonBox->addButton(quitButton,
QDialogButtonBox::RejectRole);
progressDialog = new
QProgressDialog(this);
connect(urlLineEdit, SIGNAL(textChang
ed(QString)),
   this, SLOT(enableDownloadButton())
);
connect(&qnam, SIGNAL(authentication
Required(QNetworkReply*,QAuthenticat
or*)),
   this, SLOT(slotAuthenticationRequi
red(QNetworkReply*,QAuthenticator*)))
;
#ifndef QT_NO_OPENSSL
   connect(&qnam, SIGNAL(sslErrors(QN
etworkReply*,QList<QSslError>)),
        this, SLOT(sslErrors(QNetwork
Reply*,QList<QSslError>)));
#endif
   connect(progressDialog,
SIGNAL(canceled()), this,
SLOT(cancelDownload()));
   connect(downloadButton,
SIGNAL(clicked()), this,
SLOT(downloadFile()));
   connect(quitButton,
SIGNAL(clicked()), this,
SLOT(close()));
```

Additional HTTP protocols related to signal and slots are set up in startRequest().

@code excerpt

```
 void HttpWindow::startRequest(QUrl
url)
 {
    reply = qnam.get(QNetworkReque
st(url));
    connect(reply,
SIGNAL(finished()),
            this,
SLOT(httpFinished()));
    connect(reply,
SIGNAL(readyRead()),
            this,
SLOT(httpReadyRead()));
    connect(reply, SIGNAL(downloadP
rogress(qint64,qint64)),
            this, SLOT(updateDataRe
adProgress(qint64,qint64)));
 }
```

Now let's look at the downloadFile() slot. We are reading url into QString as fileName. If the file name is not given, we automatically set it to index.html (default index file for most web servers). We are also looking for standard exceptions such as file exists, unable to save file etc. Lastly, we are also configuring progress bar to show the download progress.

@code snippet

```
 void HttpWindow::downloadFile()
 {
    url = urlLineEdit->text();

    QFileInfo fileInfo(url.path());
    QString fileName = fileInfo.
fileName();
    if (fileName.isEmpty())
        fileName = "index.html";

    if (QFile::exists(fileName)) {
        if (QMessageBox::
question(this, tr("HTTP"),
    tr("There already exists a file
called %1 in "
       "the current directory.
Overwrite?").arg(fileName),
QMessageBox::Yes|QMessageBox::No,
QMessageBox::No)
            == QMessageBox::No)
            return;
        QFile::remove(fileName);
    }

    file = new QFile(fileName);
    if (!file->open(QIODevice::
WriteOnly)) {
        QMessageBox::
information(this, tr("HTTP"),

 tr("Unable to save the file %1:
%2.")
    .arg(fileName).arg(file-
>errorString()));
        delete file;
        file = 0;
        return;
    }

    progressDialog->setWindowTitle(
tr("HTTP"));
    progressDialog->setLabelText(tr
("Downloading %1.").arg(fileName));
    downloadButton-
>setEnabled(false);
    // schedule the request
    httpRequestAborted = false;
    startRequest(url);
 }
```

In case the download is cancelled in the middle, httpRequestAborted is set to true and downloadButton is enabled again.

@code excerpt

```
 void HttpWindow::cancelDownload()
 {
    statusLabel-
>setText(tr("Download canceled."));
    httpRequestAborted = true;
```

```
    reply->abort();
    downloadButton-
>setEnabled(true);
 }
```

httpFinished() is called when the current HTTP request is finished. There are three ways in which an HTTP request can finish: (1) the download did not happen; (2) you were redirected to a new site for the download (for example If you enter http://linuxuser.co.uk as the address, the web server will redirect you to http://www.linuxuser.co.uk); (3) the download completed successfully.

We will be handling all three cases in this method. In the first case we'll just return the error text. In the second case we'll call the startRequest() with the returned URL. In the third case we'll show the message 'the downloaded completed successfully', re-enabling the download button for another download.

@code excerpt

```
 void HttpWindow::httpFinished()
 {
     if (httpRequestAborted) {
         if (file) {
             file->close();
             file->remove();
             delete file;
             file = 0;
         }
         reply->deleteLater();
         progressDialog->hide();
         return;
     }
     progressDialog->hide();
     file->flush();
     file->close();
     QVariant redirectionTarget =
reply->attribute(QNetworkRequest::
RedirectionTargetAttribute);
     if (reply->error()) {
         file->remove();
         QMessageBox::
information(this, tr("HTTP"),
 tr("Download failed: %1.")
 .arg(reply->errorString()));
         downloadButton-
>setEnabled(true);
     } else if (!redirectionTarget.
isNull()) {
         QUrl newUrl = url.resolved(
redirectionTarget.toUrl());
         if (QMessageBox::
question(this, tr("HTTP"),
 tr("Redirect to %1 ?").arg(newUrl.
toString()),
 QMessageBox::Yes | QMessageBox::No)
== QMessageBox::Yes) {
             url = newUrl;
             reply->deleteLater();
             file->open(QIODevice::
WriteOnly);
             file->resize(0);
             startRequest(url);
             return;
         }
     } else {
         QString fileName = QFileInf
o(QUrl(urlLineEdit->text()).path()).
fileName();
         statusLabel-
>setText(tr("Downloaded %1 to
current directory.").arg(fileName));
         downloadButton-
>setEnabled(true);
     }
     reply->deleteLater();
     reply = 0;
     delete file;
     file = 0;
 }
```

slotAuthenticationRequired() is called when the called URL is password-protected. It is important to note that this method is compatible with .htaccess-based password protection.

@code excerpt

```
void HttpWindow::slotAuthenticationR
equired(QNetworkReply*,QAuthenticato
r *authenticator)
 {
     QDialog dlg;
     Ui::Dialog ui;
     ui.setupUi(&dlg);
     dlg.adjustSize();
     ui.siteDescription-
>setText(tr("%1 at %2").
arg(authenticator->realm()).arg(url.
host()));
// Did the URL have information?
Fill the UI
// This is only relevant if the URL-
supplied credentials were wrong
     ui.userEdit->setText(url.
userName());
     ui.passwordEdit->setText(url.
password());
     if (dlg.exec() == QDialog::
Accepted) {
         authenticator->setUser(ui.
userEdit->text());
         authenticator-
>setPassword(ui.passwordEdit-
>text());
     }
 }
```

sslErrors() handles any SSL-related errors. We do not go into detecting the exact SSL error; rather we are just detecting if there is an error and the error details.

@code excerpts

```
void HttpWindow::sslErrors(QNetw
orkReply*,const QList<QSslError>
&errors)
 {
     QString errorString;
     foreach (const QSslError
&error, errors) {
         if (!errorString.isEmpty())
             errorString += ", ";
         errorString += error.
errorString();
     }
     if (QMessageBox::warning(this,
tr("HTTP"),

 tr("One or more SSL errors has
occurred: %1").arg(errorString),

 QMessageBox::Ignore | QMessageBox::
Abort) == QMessageBox::Ignore) {
         reply->ignoreSslErrors();
     }
 }
```

Building and executing

Copy the full source code from the **linuxuser.co.uk** article of the same name. In this section we will be creating the project file (.pro), building the project and testing it.

Copy all the source files into a directory and then run the following command to generate the initial pro file.

```
$ qmake -project
```

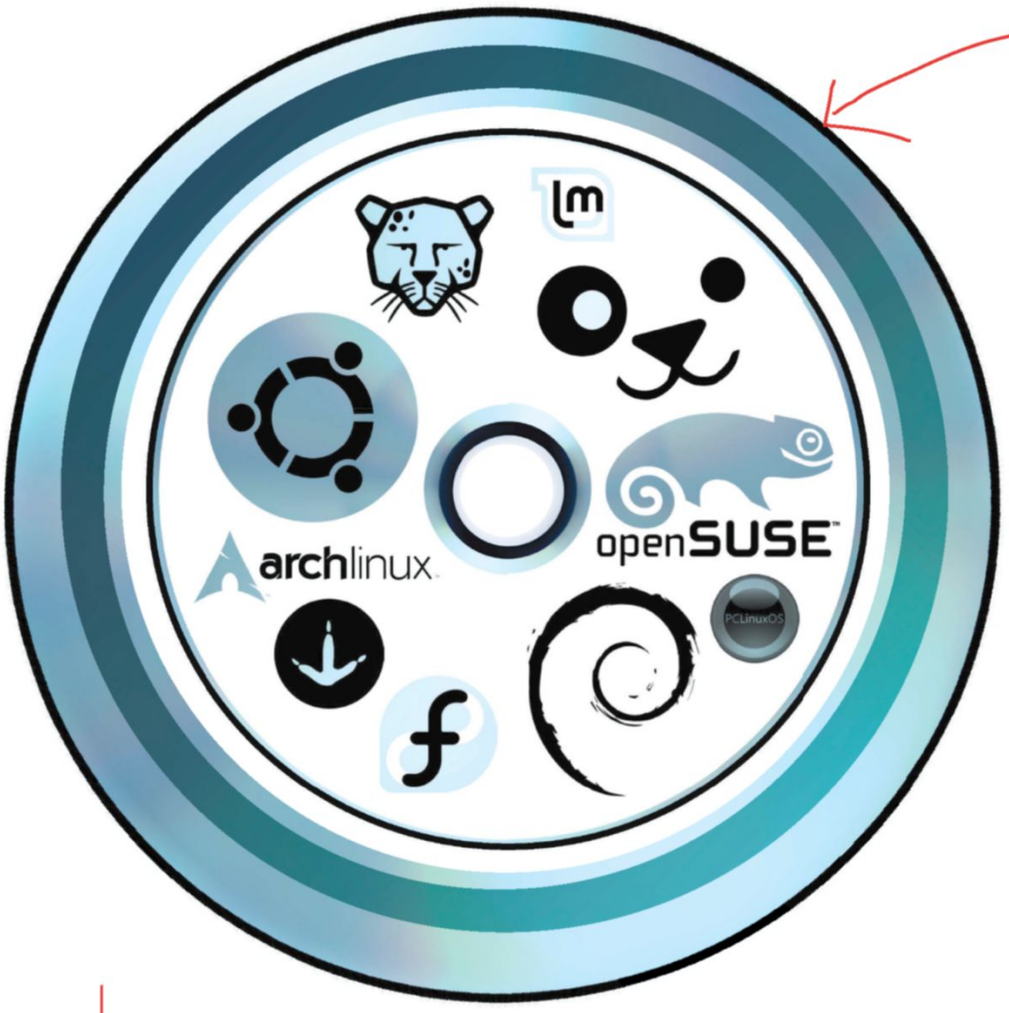
You will have the <directoryname>.pro file created in the directory. Open this file. You will notice that qmake is intelligent enough to correctly set up the HEADERS, FORMS, and SOURCES for you. Since we are using the QtNetwork module in this project, we will need to add required directive in the pro file.

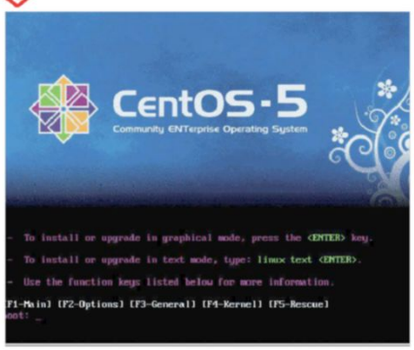
@code:<directory>.pro

```
TEMPLATE = app
TARGET =
DEPENDPATH += .
INCLUDEPATH += .
QT += network

# Input
HEADERS += httpwindow.h
FORMS += authenticationdialog.ui
SOURCES += httpwindow.cpp main.cpp
```

Essential sof

The best open source distrobutions and projects available today





"The growing popularity of distributions like Linux Mint and Pardus, could mean that Ubuntu's days at the top of the pile are numbered"


tware
archlinux
openSUSE
PCLinuxOS

10 greatest Linux distros

A lot can change in a hundred issues. In the time we've been providing you with the latest and greatest from the world of Linux in Linux User magazine, distributions have risen and fallen – and it's time we took stock.

To celebrate our 100th issue milestone, we've compiled what we believe is the definitive list of Linux distributions for the modern open source fan, by trying them first-hand as well as getting the inside scoop from community members and developers. Whether you're after something that runs on lightweight hardware, has a wide community or a slick user interface, you'll find that our rundown of the top ten Linux distributions has something for you.

A Linux distribution is a personal thing, however. While we'd like to think that our top ten list is beyond reproach, you've almost certainly got ideas of your own. Perhaps you think we've been unfair to some of the lower-ranked distributions, or you're disappointed that your favourite distribution is nowhere to be seen? If so, let us know! We're always open to feedback, and we'd love to hear your thoughts on what makes a great Linux distribution. If we've missed a particularly impressive distribution, we'll be sure to give it some love in a later issue – so please don't be shy to put your opinions forward over at LinuxUser.co.uk.

Pardus Linux

Popular in its native Turkey, Pardus is starting to make waves elsewhere

Pros
A good installer, excellent package management system, and numerous tweaks and improvements make Pardus great

Cons
Official software repositories currently have a limited selection of packages, but this is improving

Pardus has taken its home nation of Turkey by storm – to the point where it's the de facto standard for government and military use.

With the release of more internationally-friendly versions, it's starting to turn heads elsewhere in the world – and the Pardus 2011 release is the group's most impressive achievement yet.

John Alatalo, a Pardus advocate from Sweden, told us that the main draw for Pardus was its simplicity and polish. "It's the most professional and polished of the distros I have tried," he enthused, having switched from Ubuntu in 2007. "It just needs more exposure."

"Pardus includes nearly all the software I need," agreed Pardus fan Zeki Bildirici. "I just finished writing my master's thesis on Pardus, using the default software: PDF reader, office suite, citation tools. Everything worked fine. This was what I needed... and it was all covered by Pardus."

In review
ISSUE 95

Shaping up to be one of the most exciting distro releases of the year. It's a must-try distribution for distro hoppers, but needs more software packages in its repositories to gain traction with regular Linux users.

■ The Pardus 2011 live CD comes in a selection of languages

Key features

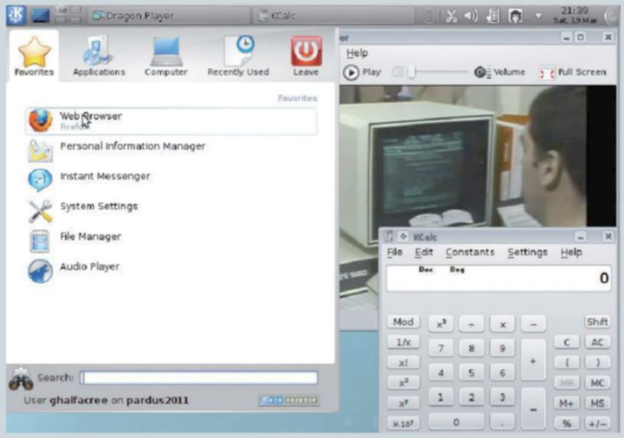
1 **The PiSi package manager** supports ratings, screenshots, and delta packages for fast updating.

2 A better than average installer – **YALI, or Yet Another Linux Installer** – is a dream to work with, if a trifle slow.

3 **The Kaptan configuration tool**, launched on first login, makes desktop customisation easy for beginners.

4 **The KDE-based interface** is clean and tidy, yet with plenty of scope for customisation.

5 **Many proprietary drivers and packages**, such as Java and Flash, are installed as standard.



■ Pardus 2011 shown running a calculator, playing a video and displaying the application menu

The facts

PARDUS WAS FIRST released in 2005 as a project of the Turkish National Research Institute of Electronics and Cryptology.

IT WAS ORIGINALLY designed as a native Turkish fork of Gentoo, but has since developed as an international distribution in its own right.

'PARDUS' IS THE Latin name for the Anatolian leopard, an image of which is used as the distribution's logo.

Puppy Linux

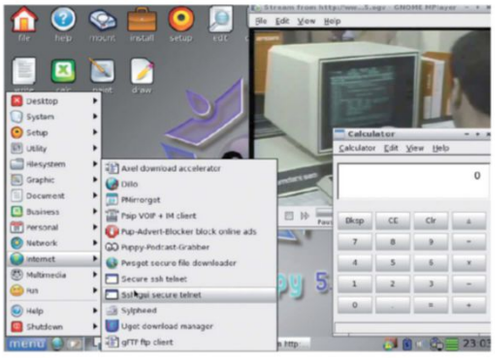
Designed for performance, it's a popular choice for lower-spec systems

The facts

PUPPY WAS ORIGINALLY developed by Barry Kauler and released in 2003, and is currently driven by Larry Short, Mick Amadio and ~~community members.~~

BUILT FROM SCRATCH to run entirely from RAM, Puppy will boot on a system with as ~~little as 48MB of RAM.~~

IF WRITTEN AS A multi-session disc, the live CD can save files and preferences to the disc for persistence between sessions.



Pros
It's great for older hardware, and its simple user interface is easy to use even for beginners

Cons
The simplified JWM user interface is unlikely to be to everyone's taste, and updating the repositories is slow

Despite having been around since 2003, Puppy is still relatively unknown in the mainstream Linux world. It's a shame, because there's a lot of power in this cuddly little distro.

Designed to load itself entirely into RAM, Puppy Linux is tiny and fast – and is a perfect choice for breathing new life into outdated hardware or creating diskless workstations.

Puppy developer Raffy Mananghaya told us that the community is key to the distribution's success. "Barry Kauler became surrounded by friendly and helpful users and devs that made the friendliness of the Puppy community legendary

PCLinuxOS

Originally a set of packages for Mandrake, it's now a well-regarded distro

Pros
A good-looking distro with some excellent software and great support material

Cons
Poor naming conventions, confusing choices, and difficult package management may put users off

From its roots as a set of custom RPM packages for Mandrake to its current incarnation as a complete Linux distro, Bill Reynolds's PCLinuxOS has grown in leaps and bounds.

As with Puppy Linux, it includes key features that make it a great live CD distro, including the ability to quickly create custom spin-off variants via a built-in script, and the option to copy the entire CD contents into RAM for diskless use.

While PCLinuxOS uses KDE as its default desktop, the community has produced spins for almost any requirement – including lightweight versions using the LXDE and Xfce desktops, along with a heavier GNOME version.

PCLinuxOS's community is small but tight-knit, meeting regularly. Looking for quotes, we approached project lead Bill Reynolds, who joked, "Unfortunately, we can't come up with anyone sane enough to talk to a reporter!"

In review
ISSUE 91

PCLinuxOS is packed with some excellent software choices and a decent array of help and support options. However, some of the design decisions completely spoil any hope of it properly catering for beginners.

■ The PCLOS live CD includes the option to run entirely from RAM for diskless use

■ The KDE interface is well laid out, offering quick access to common features

Key features

1 **The 'mylivecd' script** takes a snapshot of the current environment, making the creation of custom live CDs simple.

2 **The APT-based repository** includes an impressive 12,000 RPM packages covering most common requirements.

3 Like the smaller Puppy Linux, the PCLOS **live CD can be copied into RAM** for diskless use.

4 PCLOS includes **impressive localisation features**, with over 85 languages supported as standard.

5 While **the default version uses KDE**, spins with Enlightenment, Xfce, LXDE, GNOME Zen Mini, Openbox and GNOME are available.

The facts

PCLINUXOS, ALSO known as PCLOS, was born out of Mandrake Linux by Bill 'Texstar' Reynolds in 2003.

THE 'MYLIVECD' script allows snapshots of a live CD environment to be taken, making customisation easy.

UNLIKE ITS MANDRAKE roots, PCLOS uses the APT package management system via Synaptic.

– and the distro-hoppers attest to this."

Puppy fan and Linux system expert Igor Ljubuncic explained his love of the distribution in terms of its speed and flexibility. "I was instantly hooked when I tried it," he told us. "This was a small yet highly versatile distribution, with tons of great programs and utilities. Later versions only got better."

In review
ISSUE 90

Despite substantial changes under the hood, the new version remains lightweight, fast and user-friendly. Although the Puppy Linux 5.0 desktop has not changed much, it does sport a few improvements.

■ The default Puppy desktop is cluttered but friendly, and includes plenty of packages

Key features

1 Designed to run entirely in RAM, Puppy **can be used to boot diskless workstations.**

2 **Built-in write-caching technology** means that Puppy won't excessively write to flash storage devices if used as a live USB.

3 Puppy is **one of the smallest distributions** to include a full graphics user interface.

4 Despite its small size, Puppy **ships with almost every package** that a standard desktop user could need.

5 The Puppy Package Manager can **install packages from other distros** as well as Puppy native packages.

Arch Linux

Not the easiest distribution around, but one of the most flexible and powerful

Pros
Arch is fast, flexible and allows those with the know-how to create a lightweight, custom platform tailored to their requirements

Cons
If you're not a Linux pro, you might find the lack of GUI and relatively high learning curve a struggle

Since its creation in 2002, Arch Linux has stayed true to its roots as a minimalist distribution designed for the more knowledgeable Linux user - and while it's not the most user-friendly distribution around, it is one of the most flexible.

Current project lead Aaron Griffin believes that Arch does well not to sacrifice itself at the altar of user-friendliness. "Arch is lightweight and simple, like clay," he explains, "able to be moulded by the user as they choose. It is better to be technically elegant with a higher learning curve than to be easy to use and technically crap."

Despite not including a graphical user interface by default, Arch makes it into our top ten listing for one simple reason: it's one of the most flexible distributions out there, able to be shaped into whatever kind of platform a user requires.

■ While most configuration is done at the Terminal, packages like Wakka provide a graphical interface to package management

■ Arch Linux might not include a GUI by default, but it's still available as a live CD

Key features

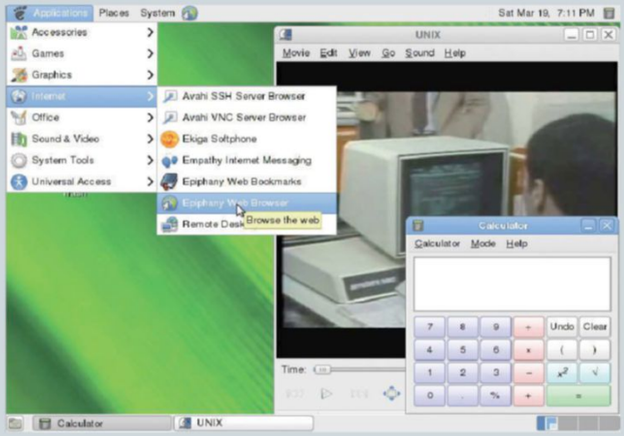
1 Arch is a completely independent distribution, designed to offer **a minimalist platform for customisation.**

2 **A custom package manager** called Pacman takes care of both binary packages and package builds.

3 While its minimalist approach makes it popular for servers, Arch has **a growing following of desktop users.**

4 **A 'rolling release' update cycle** means that an Arch system can always be updated, and never gets retired.

5 **Most configuration is done at the shell,** although third-party graphical interfaces for tasks like package management exist.



■ Arch Linux with GNOME running a calculator, playing a video and displaying the application menu

The facts

ARCH LINUX WAS created as a minimalist distribution in 2002 by Judd Vinet, and is currently led by Aaron Griffin.

ARCH IS DESIGNED AS THE basis for a customised platform, and is the only distro in our top ten not to include a default graphical user interface.

THE DISTRIBUTION'S fans take pride in the fact that Arch isn't the most user-friendly distribution out there.

Debian

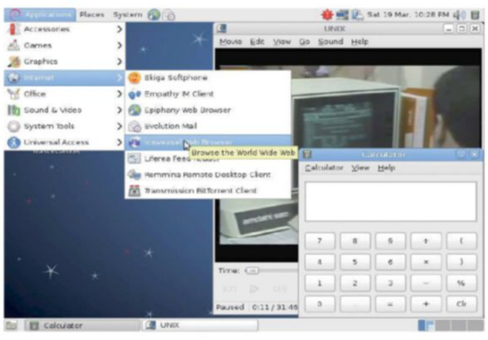
One of the oldest distributions in our list, Debian is the father to many popular distros like Ubuntu

The facts

DEBIAN GETS ITS name from project founder Ian Murdock and his then-girlfriend Debra Lynn – 'Deb-Ian'.

FIRST RELEASED in 1994, Debian is one of the oldest distributions around and has over a hundred spin-offs.

DEBIAN INCLUDES its own set of guidelines for developers: the Debian Social Contract and the Free Software Guidelines.



Pros
A clear focus to free software – despite the existence of a non-free repository – gives Debian the moral high ground

Cons
The distribution feels a little 'outdated' compared to some of its derivatives

While Debian's long release cycle has resulted in the project receiving a certain amount of negativity – directly resulting in the formation of the Ubuntu project in 2004 – there's no denying that this distribution has had more impact on the Linux world than any other.

At the last count, there were around 130 distributions that owe their existence to Debian, and while few have attained the same level of success as their parent distribution, the sheer volume is indicative of the popularity of Debian as a platform.

Sabayon Linux

One of the newer distributions in our list, Sabayon is gathering impressive steam

Pros
Sabayon inherits its flexibility from its Gentoo roots, but is significantly more welcoming to new Linux users

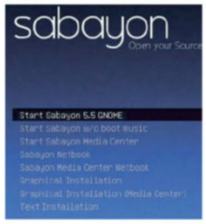
Cons
Sabayon has higher system requirements than most, and comes on a live DVD rather than a CD

Born out of Gentoo back in 2005, Sabayon is one of the newer distros to make it into our top ten – but it has a cult following who appreciate its flexibility combined with its user-friendly approach.

While Gentoo is popular, even its biggest fans would admit that the source-based distribution isn't welcoming to newcomers. That didn't put project founder Fabio Erculiani off, however. "I simply believed in the impossible," he explained. "Many people told me, it's impossible to succeed with a Gentoo-based distribution, but I showed them they were wrong."

That's an opinion that is mirrored by the distro's user base. "I think I found the best out there in Sabayon," claimed convert Dennis Accardo. "I love the way they give you the basic system and let you build it to custom proportions."

With a rolling release schedule and both binary and source-based package installation, Sabayon is a friendly alternative to Arch or Gentoo.



■ The Sabayon live DVD includes an optional full-length song as its boot sound

■ The built-in package manager is Entropy, but Portage is also available

■ Sabayon Linux 5.5 running a calculator, playing a video and displaying the application menu

The facts

SABAYON WAS THE first distribution to offer ISO images that were updated daily, although public updates are weekly.

THE DISTRIBUTION IS based on the same core components as Gentoo Linux, with added tools for automatic 3D hardware detection and configuration.

DESPITE INHERITING Portage from its Gentoo roots, Sabayon includes its own custom package manager, Entropy.

Key features

1 Despite being designed for ease of use, Sabayon hasn't lost any of the **flexibility of its Gentoo roots**.

2 Package installation from binaries or source is **fully supported via Portage and Entropy**.

3 **Proprietary codecs and hardware drivers are included as standard**, making it easy to get the system started.

4 Hardware support in Sabayon is second to none, with almost **all hardware working out of the box**.

5 As with Arch, a rolling release schedule ensures users **never get left behind** on outdated versions.

The project is currently undergoing a metamorphosis, including support for the BSD kernel alongside Linux and plans to offer the Hurd microkernel as a third option – moves that other distros are likely to watch with interest.

"Debian is committed to 100% free software," project lead Stefano Zacchiroli asserts – a focus that has directly resulted in a disagreement with the Mozilla Foundation, resulting in spin-off browser and email client projects Iceweasel and Icedove.

"Debian is one of the very few vendors – if not the only one – that is both very relevant and, thanks to its independence, can afford to take free software's side. No commercial urgency can force Debian to negotiate on that."

■ As with other distributions, Debian provides a live CD for testing

Key features

1 Debian's main focus is on **truly free software**, offering non-free packages only under sufferance.

2 Uses the popular **APT package management system**, with Synaptic for GUI users.

3 Squeeze, the most recent release, includes **an optional BSD kernel** – and future releases will feature the Hurd microkernel.

4 One of the oldest-running distributions, with a **long and distinguished history**.

5 Directly responsible for the creation of approximately **130 spin-off distributions**.

Linux Mint

This Ubuntu derivative is really turning heads with its modern twist on the distribution

Pros
Linux Mint is incredibly attractive, with a beautiful theme and icon set and a great launcher known as MintMenu

Cons
The bundled software set is a bit conservative, and it has few additional features beyond Ubuntu

Ubuntu spin-off Linux Mint was founded to achieve the impossible: take a distro already well known for being welcoming to newcomers and make it even easier and friendlier. By all accounts, it's something the project has succeeded in doing.

Linux Mint has a looser approach to free software ideals than its grandparent distro Debian. While FLOSS software is the focus, non-free proprietary packages are included if it makes life easier – so features like Adobe Flash and H.264 playback are included as standard.

"Mint is the second most popular distribution on the home desktop market at the moment," project founder Clement Lefebvre told us. "Its main principles revolve around asking the minimum number of questions and providing sane, good defaults – like in Mac OS – while allowing the user to change just about anything, should he feel like it."

In review
ISSUE 96

While Linux Mint 10 doesn't introduce any new features, a slew of tiny yet important tweaks improve the overall usability of the system and make it the best alternative to Ubuntu.

■ Linux Mint aims to be the most welcoming distro to newcomers

Key features

1 **The Mint-X theme and Faenza-based icon set** make Linux Mint one of the most attractive distros around.

2 **The MintMenu launcher** highlights newly installed packages, and includes integrated search functionality.

3 A comprehensive first-boot welcome screen makes **installation of proprietary codecs a breeze.**

4 **Linux Mint's Software Manager** is a distinct improvement over a bare-bones Synaptic install.

5 Echoing Ubuntu's release cycle, **Linux Mint is kept up to date** with six-monthly releases.

■ Linux Mint 10 'Julia' running a calculator, playing a video and displaying the application menu

The facts

LINUX MINT FOCUSES on ease of use for newcomers to Linux, yet it is still highly customisable.

PROJECT FOUNDER Clement Lefebvre aims to create a distribution for both novices and experienced Linux users.

ALTHOUGH PRIMARILY based on Ubuntu, a direct Debian derivative – Linux Mint Debian Edition – is also available.

Fedora

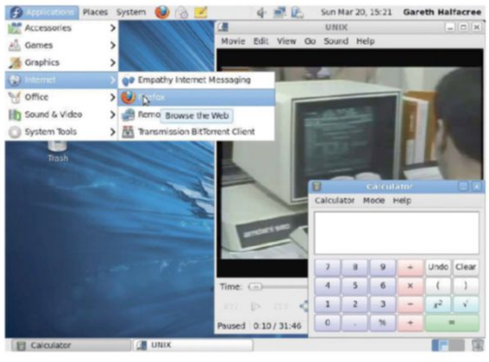
Red Hat derivative Fedora is still an excellent choice of distribution for the more technically minded

The facts

FIRST RELEASED IN 2003, Fedora Core – as it was then known – was to take over from the retired Red Hat Linux.

THE DISTRIBUTION grew out of a volunteer project providing additional software for Red Hat Linux.

FEDORA IS SO NAMED for the type of hat worn by the 'Shadowman' in Red Hat's logo.



■ Fedora 14 GNOME spin with calculator, video and app menu

Pros
Powerful encryption capabilities, SELinux support and a great firewall make it ideal for the security conscious

Cons
Many areas are still a bit unfriendly to recommend Fedora to Linux newcomers

When the long-running Red Hat Linux distribution was due to retire – as the company behind it sought to focus on its Red Hat Enterprise Linux commercial distribution – fans gave it a new lease of life in the form of Fedora Core.

Now known simply as Fedora, it's an increasingly popular distribution.

While it lacks the user-friendliness of distributions like Linux Mint, Fedora has a cult following – and a focus on security in recent releases, with features like full-disk encryption and default-on SELinux access

openSUSE

Primarily sponsored by Novell, openSUSE has a strong following and great community

Pros
Tumbleweed offers a way to keep on the bleeding edge between releases, and openSUSE's KDE spin is undeniably pretty

Cons
Certain high-profile packages like WordPress and Dropbox aren't available in the default repositories

Originating back in 1994, openSUSE is one of the oldest distros to make it into our top ten. Unlike Debian, however, its development team has worked hard to keep on top of changes in visual styles, giving the distribution some KDE 4.6 glitz in the most recent release.

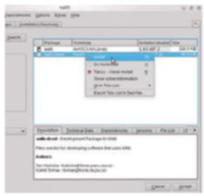
Perhaps the biggest draw is the sheer volume of bleeding-edge packages on offer: version 11.4 ships with KDE 4.6, LibreOffice 3.3.1 and even a beta release of Firefox 4. A rolling-release update repository, Tumbleweed, keeps everything updated, too.

"OpenSUSE is a tool that lets me do most things without having to get my hands too dirty at the command line," project member Andrew Wafaa told us.

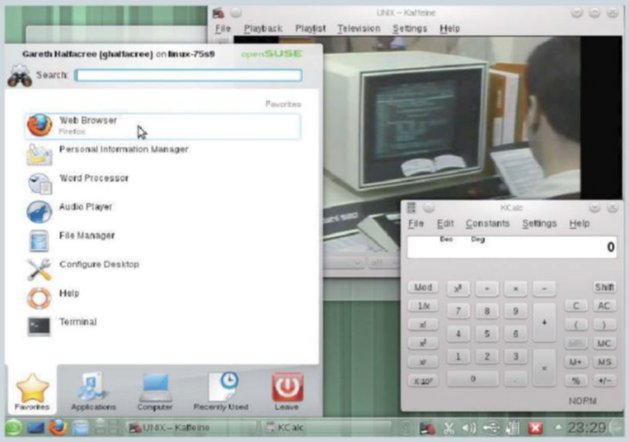
"Why do I use openSUSE? Because of all the power behind a few funny words like Zypper, KIWI, WebYAsT, LibreOffice and more," said openSUSE chairperson Alan Clark.

In review
ISSUE 99

11.4 mostly boils down to newer software releases, which is not irrelevant with openSUSE's eight-month release cycle. If you find this cycle too long, Tumbleweed can definitely help you out.



■ The YaST2 package manager isn't the prettiest around, but has plenty of power



■ OpenSUSE 11.4 running a calculator, playing a video and displaying the application menu

Key features

1 **Tumbleweed** makes it easy to stay on the bleeding edge, providing users with regularly updated packages.

2 **The KDE Plasma interface** is one of the best we've seen, with great visuals.

3 Advanced users are backed up with a great toolchain, including **a Build Service and the SUSE Studio site**.

4 32-bit users have support for **4GB of RAM straight out of the box**, with no PAE installation required.

5 Customisation is easy, with the ability to roll your own **live CDs with custom content**.

The facts

ORIGINALLY A restricted-release commercial distribution, openSUSE is one of the oldest distros around.

FEATURES A GREEN gecko, known as the Geeko, as its mascot. It was the result of a competition and its design has evolved over the years.

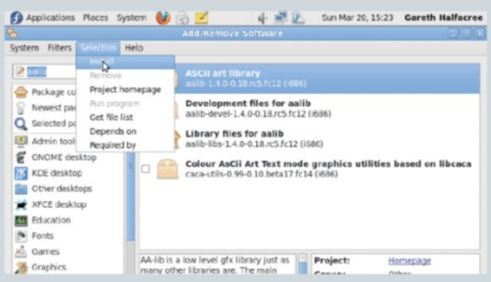
THE OPENSUSE community is backed by impressive toolkits, including the popular Build Service site.

control has helped push that along.

"The main reason I go for Fedora has always been its security," explained Fedora fan Kereith Foster in response to the Fedora 14 review we ran back in issue 94. "On the whole I am impressed with it, and will be using it for the foreseeable future."

While Fedora fans admit that the distribution can require some 'tweaking' before it's fully working, they maintain it's worth the effort – and point to a knowledgeable community who are more than happy to lend a hand to a new user as one of the distribution's key features.

The impending release of Fedora 15, which features GNOME 3 and the GNOME Shell, will likely have a major impact on the distribution's ongoing popularity.



■ Fedora's default package manager enables you to add and remove software with ease

In review
ISSUE 94

There's a lot to recommend Fedora 14, from its well thought out security features to the wide variety of development tools available. But there are some issues with outdated software and SELinux.

Key features

1 Fedora's installer includes **the option to create a totally encrypted installation**.

2 SELinux, **the NSA-approved access control system**, is included with Fedora and enabled by default.

3 Webpage rendering is given a boost thanks to **the libjpeg-turbo library**, included since Fedora 14.

4 **The focus on GNOME** has not stopped the KDE spin of Fedora gaining popularity in recent years.

5 **Excellent developer tools** make Fedora easy to recommend for the technically minded.

Ubuntu Linux

A Debian spin-off, Ubuntu is undoubtedly the most popular desktop Linux distro around

Pros
Ubuntu is extremely popular and enjoys great support from hardware and software vendors

Cons
Many are still unconvinced by the distribution's move to Unity as a desktop interface

Mention 'Ubuntu' in the presence of a hardcore Linux user and you'll likely get a snort of derision. Much has been made about project owner Canonical's lack of support for its upstream distro Debian, and many feel that recent decisions – such as the move to home-grown interface Unity rather than GNOME Shell – are actively harmful to the Linux community.

Like it or loathe it, there's one thing you can't deny: Ubuntu is a seriously popular distribution.

Canonical, the company behind Ubuntu, is aware that the distribution doesn't have the best reputation among the wider community and is actively working to fix that. "We have begun a project to test out a new approach to the problem," Canonical's chief technical officer Matt Zimmerman explained. "DEX is a joint task force where developers from Debian and its derivatives work together on this common goal. As a pilot project, we've established an Ubuntu DEX Team focused on merging code from Ubuntu into Debian."

This team will help ensure that patches developed for Ubuntu get fed back into Debian – addressing one of the major concerns that people have with the distribution.

For a user who just wants to use Linux, there's a lot to recommend Ubuntu. It has a friendly user interface, one of the largest communities around and a great selection of bundled software.

The growing popularity of competing distributions like Linux Mint – which is based on Ubuntu – and Pardus, both of which offer a similar level of ease of use, could mean that Ubuntu's days at the top of the pile are numbered, however.

In review
ISSUE 100

Check out the full review of the first beta for Ubuntu 11.04 (Natty Narwhal) on page 82 of this issue.

The facts

UBUNTU IS NAMED after the South African philosophy of 'ubuntu,' which means 'humanity towards others'.

RECENT STATISTICS suggest that Ubuntu enjoys a 50% share of the Linux desktop market, with more than 12 million users.

UBUNTU WAS BORN from the Debian distribution after concerns were raised about its long release cycles.

Key features

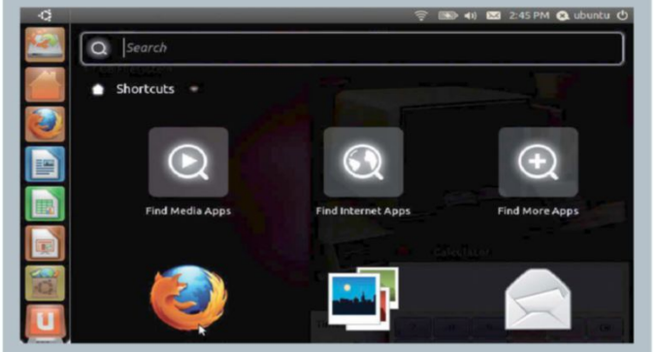
1 Ubuntu is **the most popular desktop Linux distribution** around at the moment, with over 12 million users.

2 A **'Software Center'** makes finding new packages – and even buying commercial packages – easy.

3 Ubuntu is designed to offer **a single distribution** for tablets, PCs and laptops, and netbooks.

4 **Backed commercially by Canonical**, which offers for-cost enterprise support and add-on packages.

5 Ubuntu's **popularity in the server world is growing**, with Amazon offering Ubuntu cloud implementations on EC2.



■ You can read our Ubuntu 11.04 beta review written by Dmitri Popov starting on page 82

BUBBLING UNDER

Not every distribution has what it takes to make it into our top ten, but that doesn't mean they're not worth investigating. Here's a selection of our favourites that didn't make the final cut...

PEPPERMINT

Peppermint is another Ubuntu spin-off, but this time focuses on cloud computing. When we reviewed Peppermint Ice back in issue 91, we were impressed by the simplicity of its 'Ice' applet for creating shortcuts to cloud-based apps and its speed – but it was ultimately let down by a lack of features.

MANDRIVA

It's was a hard choice between Red Hat spin-offs Fedora and Mandriva, and while Fedora made it into the top ten, its sister distribution is worth a glance too. Our review of Mandriva 2010 Spring in issue 90 praised the distribution for its modern looks and new features, but some issues mean that it loses out to user-friendly alternatives like Ubuntu and Linux Mint.

TINY CORE

While Arch exists for the techies and Puppy for those with older hardware, those who want something truly modular should look at Tiny Core. Back in issue 99, we were amazed at the distro's ability to boot into a (minimalist) desktop from a 10MB ISO. Sadly, infuriating defaults and a difficult install process kept it out of the top ten.

JOLICLOUD

Jolicloud is one of the most impressive distributions we've seen in recent times. Based around the same cloud computing concepts as Peppermint Ice, Jolicloud uses an HTML5-based interface over a customised Ubuntu to offer seamless cloud computing. Our review in issue 91 showed how far the French distribution has come since its early days, but the hiding of legacy apps and surprisingly limited social networking functionality just kept it from our top ten.

GROUP TEST

Backup supertest

Areca Backup

Areca Backup is a Java-based cross-platform package with some impressively powerful features under its belt

Areca is a powerful piece of software – and its Java-based cross-platform nature means it runs on most Linux distros, and Windows and Mac, without issue. It has impressed us enough that we have a step-by-step guide to the software over on page 36.

Installation is a breeze, with pre-compiled binaries available for both 64-bit and 32-bit versions of Linux. For a standard user, the installation archive can be extracted anywhere and the program executed; for installation for all users, a root user can stick it in the system path.

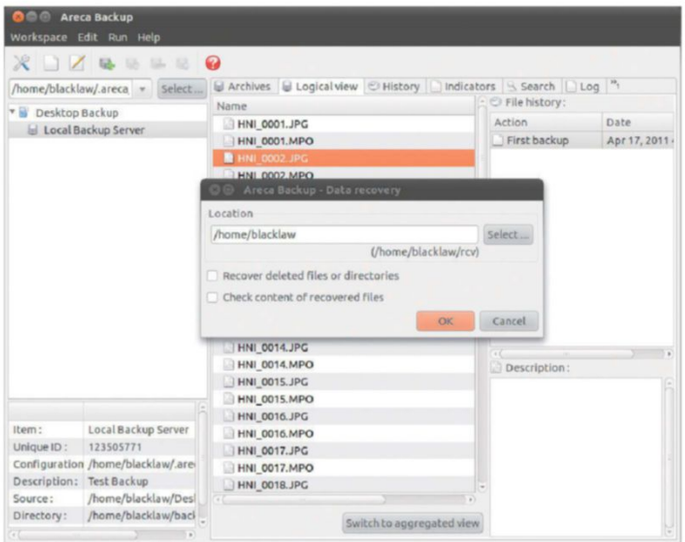
The user interface in Areca has had plenty of tweaks over the years, and provides quick and easy access to all the required features – including a comprehensive log file showing exactly what the program has been doing. It also makes it easy to walk back along different backups to find files for recovery, with a very straightforward interface for even the least experienced backup user.

Backup destinations, known as 'Targets', can be created on the local file system or on remote FTP, FTPS, or SFTP servers – although this functionality is missing a few tricks, with no easy way to add an SSH key for passwordless authentication. The entire source and destination configuration can, in fact, be a little daunting at first – although once configured, running or testing backups is pretty straightforward.

Backups can be compressed with ZIP or ZIP64 – the latter required if backing up more than 4GB to a single file – and protected with 128-bit or 256-bit AES encryption. The option to encrypt the filenames in addition to their contents adds an extra level of security to proceedings, too.

Once a backup source and destination is set, running a backup is easily managed through the GUI. The package also has the option of creating a shell script using the CLI, which performs the backup automatically. While this can be added into any scheduling system, such as cron, for automated backups, there's no interface to do this via Areca itself – it's a manual task.

We had no problems running Areca across even large directories, although the 64-bit build can't store ACLs and extended attributes. The 32-bit build doesn't have this limitation, though.



■ Entire backups or individual files can be recovered and verified against the backup

DETAILS

Installation
Single-user installation is just a question of extracting the files, but multi-user installation requires manual intervention

Features
Remote server support, compression and differential backups are all included

Flexibility
A clever plug-in structure makes it easy to add functionality to Areca

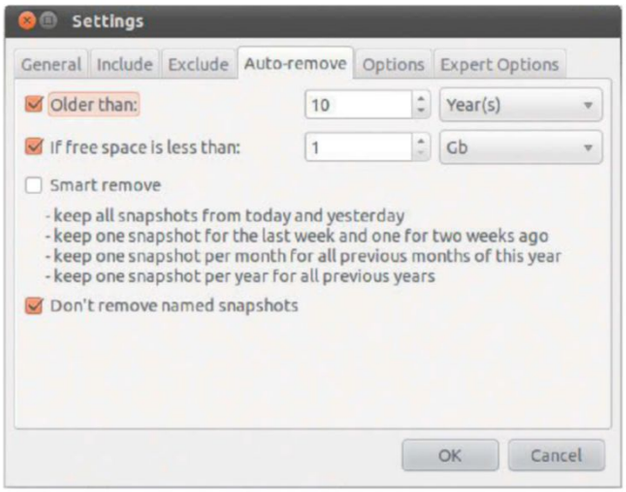
Usability
The interface is good, but 'Target' creation is complex and scheduling difficult for less technical users

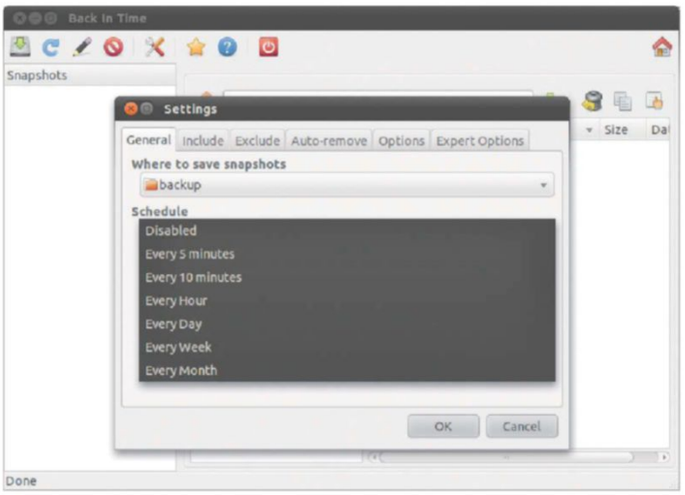
Overall
Areca is a great choice for those who need to support Linux, Windows and Mac from a single package – but configuration isn't as straightforward as it could be

More information
www.areca-backup.org

Back in Time

With interfaces for GNOME and KDE, along with Nautilus integration, Back in Time features a simple GUI





- Integrated scheduling support makes it easy to keep your Back in Time backups up to date
- Older backups can be removed (or 'Smart-removed') to save space

Unlike Areca, Back in Time is simply a front-end to other utilities – specifically, rsync, diff and cron. By tying these tools – traditionally used by system administrators to carry out backups and remote directory synchronisation – into a friendly graphical user interface, Back in Time makes it easy to configure automated backups on any Linux system.

Installation is straightforward, with both KDE and GNOME variants of the tool included in many distributions' repositories. Additional tools are also available for tying the package in to Nautilus, adding context-sensitive options to the right-click menu.

Back in Time is one of the easiest-to-use packages on test. Adding folders to its backup system is incredibly simple, while its integration with cron means that scheduling is straightforward. Backups can be scheduled to run at intervals as frequent as five minutes and, thanks to the rsync-powered file transfer, are incredibly fast to run once the initial snapshot is taken.

The ease of use, sadly, comes at a cost to its flexibility. There's no built-in option to connect to a remote server to store backups off-site, and files saved via the application aren't compressed to save space – although rsync does ensure that only changed files are copied to each subsequent snapshot.

The software's snapshotting system works well, tracking changes between each individual snapshot to provide a 'point-in-time' recovery system for changed, deleted or overwritten files. The interface makes it easy to browse through the file system and check snapshots for each file, and a differential system highlights changes between backed-up and current copies.

To prevent the uncompressed backup files from getting too large, especially in scheduled mode where backups occur without user interaction, the software can be configured to delete old snapshots – ensuring that there are always a sensible number of backups to restore from at any given time.

With a great user interface, simple configuration and the ability to integrate nicely with both GNOME and KDE, it's easy to recommend Back in Time – although more advanced users will feel limited by the lack of flexibility, while those with limited hard drive space will miss the option to compress backup files. The lack of a built-in option for connecting to remote servers is also a sad omission, as this removes an easy way to add disaster recovery to a backup schedule.

DETAILS

Installation
Back in Time is provided for both 64-bit and 32-bit distributions, and has very few strange dependencies

Features
The software performs basic backups well, but lacks advanced options such as encryption and compression

Flexibility
Back in Time is a poor choice for those who like fine-grained control over their backup procedures

Usability
The user interface is simple and configuration is quick and easy for both one-off and scheduled backups

Overall
Back in Time is excellent for newcomers or those who want a simple backup system, but lacks power in certain areas

More information
http://backintime.le-web.org/

luckyBackup

Like Back in Time, luckyBackup serves as a front-end to otherwise complex tools like rsync

For those who find the configuration of command-line tools an arduous task, luckyBackup offers another alternative. Like Back in Time, it ties in to existing command-line tools like rsync, providing a graphical user interface that works well with both GNOME and KDE; and easy scheduling options for unattended backups, powered by cron. Back in issue 90, we used luckyBackup as the basis for a step-by-step backup guide.

Installation is straightforward, with a version available in most repositories in both 32-bit and 64-bit flavours. Unlike Back in Time, however, by default luckyBackup runs purely in superuser mode. While this ensures that the software has full access to any files it needs to access, it does make it harder for a non-privileged user to back his or her files up using the software without the administrator granting such permissions.

Creating a backup is slightly less straightforward than Back in Time, largely due to a slightly cluttered and dated user interface - although still far from difficult even for newer users. Basic support for connecting to remote servers is also available, making it easy to create a backup that stores to an off-site system in case of a major disaster locally.

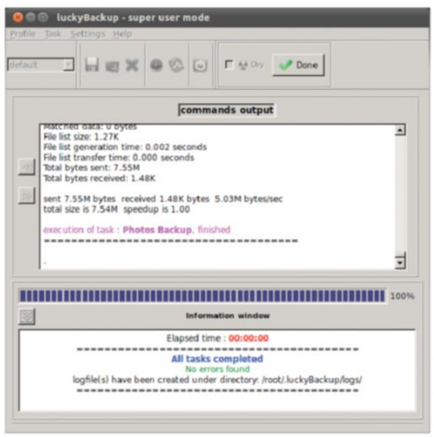
Backups are copied to their destination using rsync, ensuring that only changed files are transferred - and, as with Back in Time, point-in-time recovery is supported: multiple snapshots can be stored and any file from any snapshot restored to its former location.

Sadly, there are some limitations, including no facility for creating compressed backup files, meaning that backups of frequently changed directories can rapidly soak up disk space on the target system.

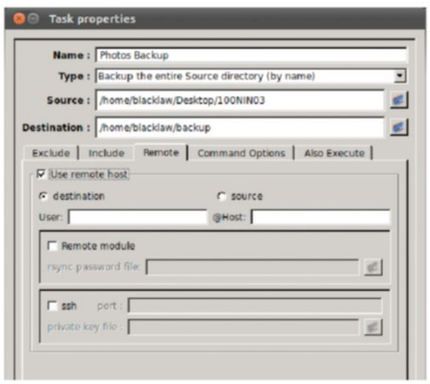
Permissions and ACLs are stored with the changed files, with the option to support NFTS and FAT target systems for backing up to external drives that will also be accessed by Windows or Mac systems.

With no encryption support, users backing up to external storage – or a remote server – will have to use a third-party application like TrueCrypt or an encrypted logical volume to ensure that their files are kept private in the case of a backup volume going missing.

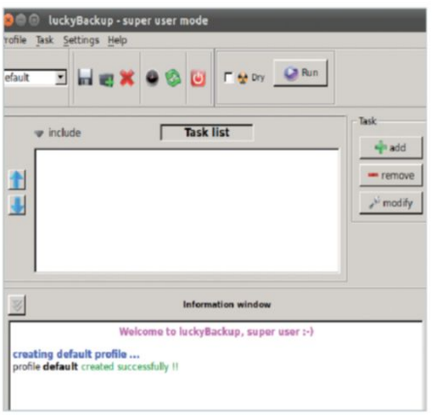
While luckyBackup is a good choice for those who need to back up to remote servers, its user interface is somewhat dated – and somewhat more complicated to navigate than the similar Back in Time.



A comprehensive log file makes luckyBackup easy to monitor



Backup jobs are labelled as 'Tasks' in luckyBackup, and remote servers are also supported



The user interface in luckyBackup could be clearer, and feels somewhat dated

DEATILS

Installation

Binaries are included for most distributions, but shortcuts are only created for privileged users by default

Features

No encryption or compression support sorely limits luckyBackup's flexibility

Flexibility

While still limited compared to Areca, luckyBackup adds remote server support to Back in Time's feature set

Usability

The user interface is slightly cluttered and dated, but most features are still easy to access

Overall

For those needing remote server support, luckyBackup is a good choice – but for local backups, Back in Time is a better choice

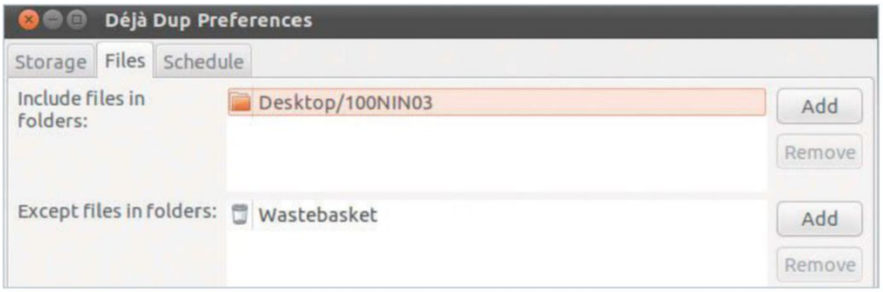
More information

http://luckybackup.sourceforge.net/

Déjà Dup

Offering robust encryption and remote server support, Déjà Dup is a powerful tool with a very simple interface

■ For novice users, Déjà Dup's simple two-button interface is hard to beat



■ While Déjà Dup's user interface is basic, it makes it easy to add files to a backup

While newcomers to Déjà Dup might find the simple interface - dominated by two buttons, one to back up and one to restore - insultingly basic, there's a lot of power hidden under the bonnet. With support for remote backups, encryption, and compression, it's a surprisingly powerful utility.

Installation of Déjà Dup is extremely straightforward: packages are included for most distributions in both 64-bit and 32-bit flavours, and use of the tool doesn't require superuser permissions.

The user interface provided with the package is remarkably straightforward, with new users being presented with just two options: restore, which accesses existing backups; and backup, which creates a new backup or updates an existing one.

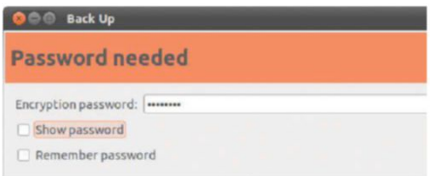
While the interface might make some dismiss the software as a bare-bones package for beginners, there's a surprising amount of power behind the two buttons. Backups created in the software can be encrypted (via the GNU Privacy Guard package, support for which is integrated into the software) and stored on remote servers. Built-in support for Amazon's S3 cloud-based storage service is also included as standard.

For those without Amazon S3 or remote server access, Déjà Dup is equally at home storing its files on a local or external storage device – and fully supports backing up and restoration of ACLs and extended file attributes.

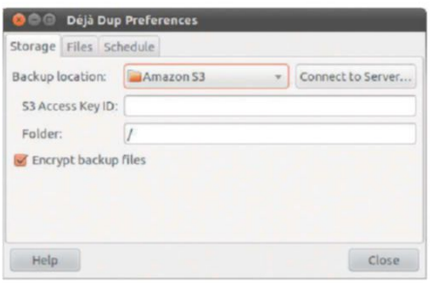
During our testing, Déjà Dup worked well at backing up our files and directories – although problems can occur during the creation of the backup archive itself, which requires temporary space on your local drive prior to the backup being uploaded to a remote server.

Sadly, the simple user interface does fall down on one important aspect: restoration. By default, Déjà Dup only provides the facility to restore an entire snapshot – either to its original location or to a new folder. There's no easy way to browse the files held within a backup, and nor is it possible to restore just a single file – unless you can remember its exact name and location, in which case it can be achieved with a command-line flag.

If you're likely to need to restore older versions of individual files, as opposed to restoring an entire missing directory in the event of a system or hard-drive failure, you would probably be better off using Areca.



■ Backups created with Déjà Dup can be encrypted via GPG for added security



■ Déjà Dup includes integrated support for Amazon S3, for cheap off-site cloud storage

DETAILS

Installation

Déjà Dup is easy to install and once it's on the system, it doesn't require superuser permissions to run

Features

Remote servers are included as a target option – including Amazon's S3 – and GPG encryption comes as standard

Flexibility

Backing up is quick and easy, but restoring an individual file from a backup is far harder than it needs to be

Usability

For most users, the simple user interface is a blessing rather than a curse – but more advanced users should look elsewhere

Overall

Déjà Dup is by far the easiest package to use out of our test suite, and includes some powerful features – but its lack of flexibility can make it a pain for users with non-standard requirements

More information
https://launchpad.net/deja-dup

GROUP TEST

Media player super-test

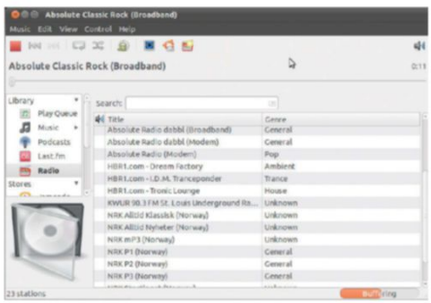
Rhythmbox

The Rhythmbox media player is the default for Ubuntu 10.10, but was it the right choice?

Rhythmbox is designed to offer integrated music management in an interface heavily influenced by Apple's iTunes, but don't let that put you off: it offers some powerful facilities that go above and beyond its roots.

Rhythmbox supports a wide variety of audio formats including – providing you've got the non-free GStreamer extensions installed – MP3, MP4, FLAC and Ogg. There's no support for video files, but Rhythmbox is an excellent music player. Integrated support for ripping, playing, and burning audio CDs is a welcome bonus, and support for iTunes-powered portable media devices as well as MTP and USB mass storage format players means that it plays well with your existing hardware.

Music management is a key feature, with the software keeping an index of all your tunes on both your system and your external devices. Automatic downloading of album art and even song lyrics is included – although for this to work, your metatags need to be accurate.

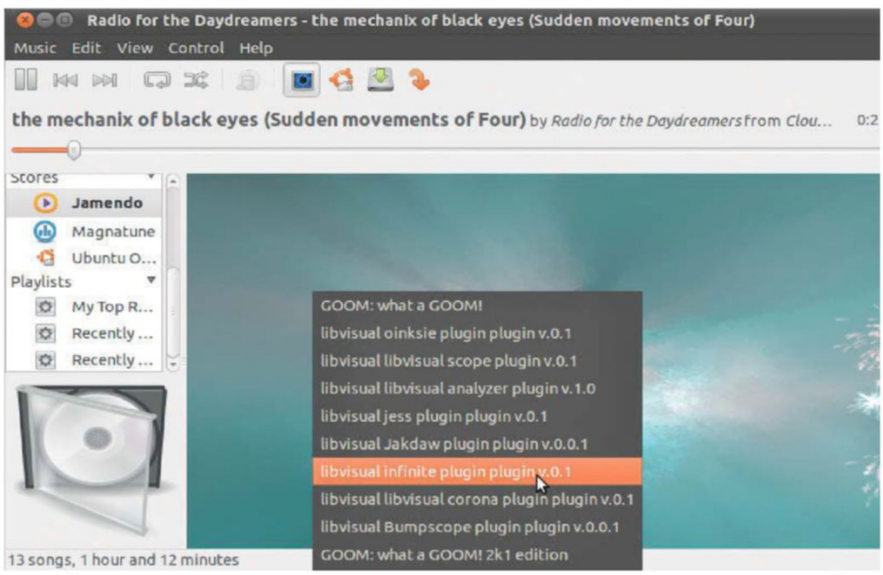


A built-in list of internet radio stations is a handy extra feature of Rhythmbox

Perhaps the most impressive feature of Rhythmbox is its support for internet services such as Jamendo and Magnatune. These web record labels offer free streaming of music and Rhythmbox makes it easy to browse the wide selection and immediately start listening to your choice of music. For the bundled version offered with Ubuntu 10.10, support is added for the Ubuntu One Music Store – making Rhythmbox even more like iTunes with the ability to browse and buy popular music in a DRM-free format.

For Last.fm users, scrobbling – where the music you play is shared with Last.fm's servers to make the streaming radio more accurate – is supported via the default API.

Rhythmbox integrates well with the GNOME desktop, offering control from the system tray and notification messaging when tracks change. If you're a KDE user, however, you'll need to install the GNOME libraries in order to run the package – in which case, you might be better off looking at the KDE-specific Amarok instead.



A wide range of visualisation plug-ins keep your eyes occupied

DETAILS

Features

Rhythmbox is an undeniably good music player with great online features, but loses points for the lack of video support

User interface

The user interface is clean and tidy, and it integrates well with GNOME

File compatibility

The GStreamer back-end gives good compatibility, but no video files are supported

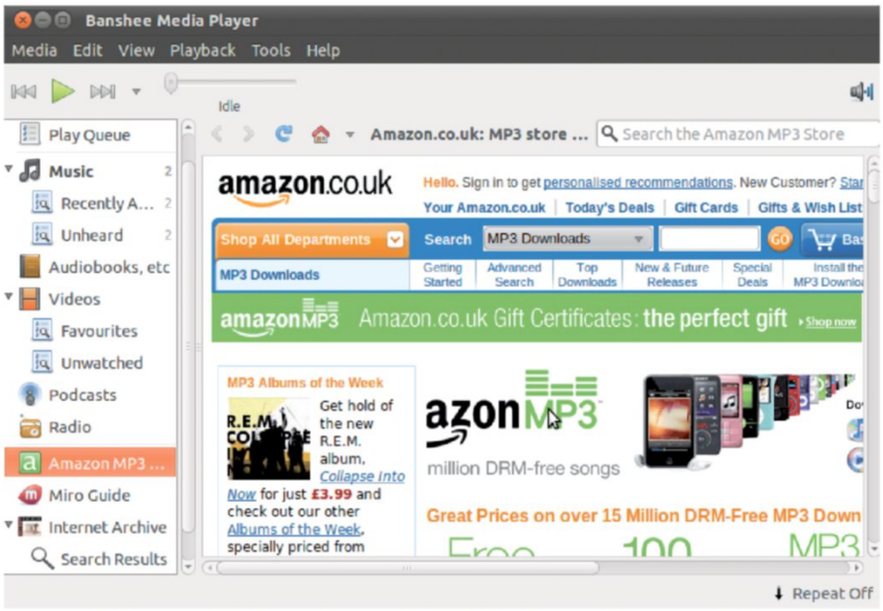
Overall

Rhythmbox is a good choice for music lovers, but you'll need a separate video player

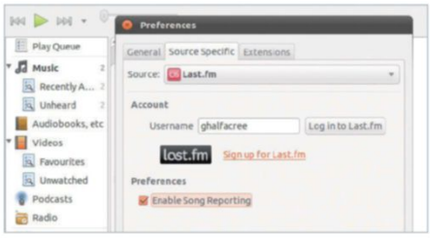
More information
www.rhythmbox.org

Banshee

Banshee aims to offer an all-in-one package for music and videos, but can it deliver?



Banshee's built-in Amazon MP3 store support is extremely handy



As with other packages on test, Banshee includes Last.fm scrobbling support

Support for content from the Internet Archive is a surprising, but welcome, addition

The Banshee media player and the Rhythmbox music player have been at odds recently, following the decision by Canonical to make the move to Banshee as the default media player in Ubuntu 11.04 and onwards. While the Rhythmbox crowd took it as an insult, the promise of a one-stop player for both music and video appears to have been too tempting to miss.

Banshee is another GStreamer-powered package, but unlike Rhythmbox it takes full advantage of the power on offer by supporting both video and audio files – and we struggled to find a format that the software didn't accept, with MP4 video and audio, MP3 audio, and Ogg video playing back perfectly during our tests. The only time we found Banshee struggling was when presented with an Adobe Flash Video file, which it refused to play no matter what we tried.

Don't think that the addition of video support means that Banshee has neglected the music side of things, however: as with Rhythmbox, support for media library management on both internal storage and external devices is fully supported. Album art is automatically downloaded for songs with accurate metadata, while USB mass storage, MTP format and Apple media devices are fully supported – along with the somewhat rare Rio Karma players.

Banshee's support for online media sources is excellent, with the Amazon MP3 store supported out of the box – one of the easiest ways to browse and buy MP3s on a 64-bit Linux system, considering that the official Amazon MP3 Downloader app only supports 32-bit Linuxes. Support for Miro is also included, along with Creative Commons-licensed content from the Internet Archive. A built-in list of internet radio stations completes the offering – along with support for downloading podcasts via RSS feeds.

An impressive feature of Banshee is the support for multiple shuffle modes – including the ability to generate a playlist based on the acoustic similarity of songs. While it doesn't always deliver the results you might expect, it's a neat trick – and certainly makes a change from the usual genre-based shuffling options on offer from other media players.

Integration with GNOME is again tight with Banshee, although other platforms are supported. If you like what Rhythmbox has to offer, Banshee is certainly worth considering as an alternative if you would like to use the same software to listen to music as you do to watch videos.

DEATILS

Features

Banshee is jam-packed with features, and has plenty to offer its users

8

User interface

The user interface is clean and tidy, with everything close at hand

7

File compatibility

The GStreamer back-end offers good compatibility, but certain formats are still a struggle

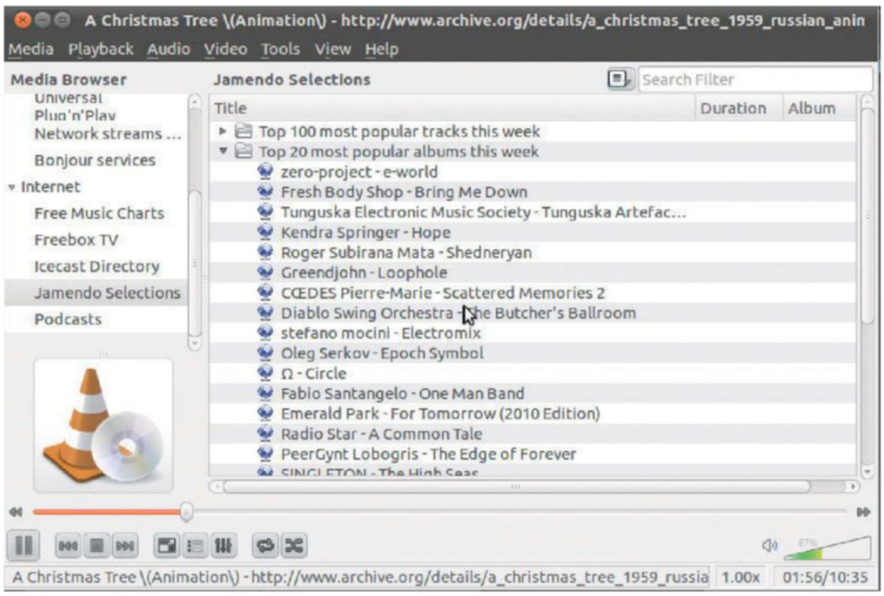
Overall

If you're looking for a media player with good music management functionality and also video support, Banshee comes highly recommended

More information
http://banshee.fm/

VLC Player

The VLC Player is a powerful piece of software, but can it hold its own against rival packages?

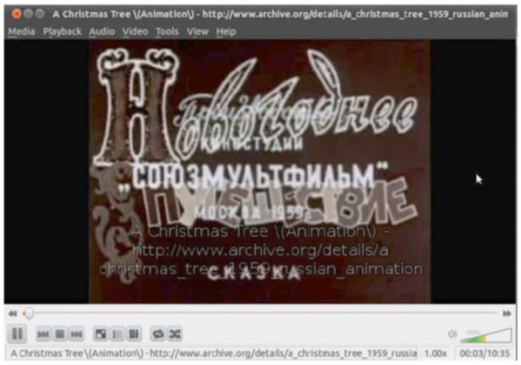


VLC includes support for network-based and internet-based media sources

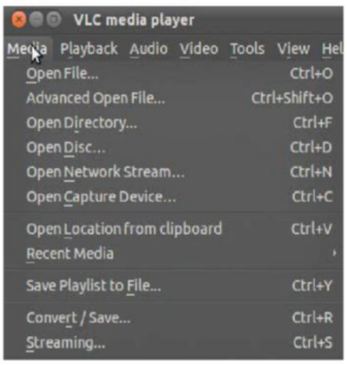
The VideoLAN project is possibly the most renowned organisation in the world of open source multimedia. Originally started as a student project at the École Centrale Paris in 1996, the project's best-known creation is VLC Player – possibly the most feature-rich media player in the world.

The first thing that strikes a new user about VLC is, unfortunately, a negative: the user interface is somewhat outdated. Taking its cues from the old Windows Media Player and similar apps, it doesn't offer anything like the friendly interface of rival packages such as Amarok or Banshee. There's little to no support for library management – although playlists are supported – and the default view is of a compact, minimalist window.

Behind its dated looks, however, lies an absolute beast of a package. While you can use VLC Player to just play back your audio and video, it also offers advanced functionality that isn't matched by anything else on test. You can access video capture devices and stream them across your network to other VLC Player clients, for example, or transcode a video file into another format automatically.



VLC Player is most at home when dealing with video content, of almost any format



The sheer flexibility of VLC Player has to be seen to be believed

DETAILS

Features

The feature set is unbelievably powerful, but lacks library management facilities

8

User interface

The user interface isn't for everyone and while skins make things better, it still needs work

7

File compatibility

There's no media player around that offers better compatibility than VLC Player

Overall

VLC Player is hands-down the most powerful package on test, but is likely to be overkill for most users

More information
www.videolan.org

File support is also a major feature of VLC Player. Unlike other players on test, the software doesn't rely on a back-end package such as GStreamer to play back its media files. Instead, it has the codecs it needs built straight into the package itself – giving it the best compatibility of any package on test, hands down. The VLC Player stormed through our test suite of audio and video files with aplomb, even playing the Adobe Flash Video that Banshee struggled to open.

Where VLC Player fails to stack up to its competition is in online support. While it will happily stream almost any audio or video source on the internet, it lacks the built-in browser and content stores of its rivals. That said, it does include a neat browser for the Jamendo service hidden away in its depths.

While there's no easy way to add such functionality into the software, there is a way to make it a bit more pleasing on the eye: skins. Available for download from the VLC Player website, the skins pack makes the software significantly more visually appealing.

Amarok

Designed specifically for KDE, can Amarok offer a native equivalent to Banshee or Rhythmbox?

Like the GNOME-centric Rhythmbox, Amarok is designed to offer KDE users an interface for their music playback needs – and the two share plenty of features, with both offering impressive web-based functionality.

The user interface in Amarok is clean and tidy, with a three-pane view that makes it quick and easy to find music from local devices, the internet and in folders. As with other packages on test, integrated podcast support is included – and it's easy to add in an RSS feed and have Amarok automatically download each new podcast as it gets released, thanks to support for the OPML podcast directory.

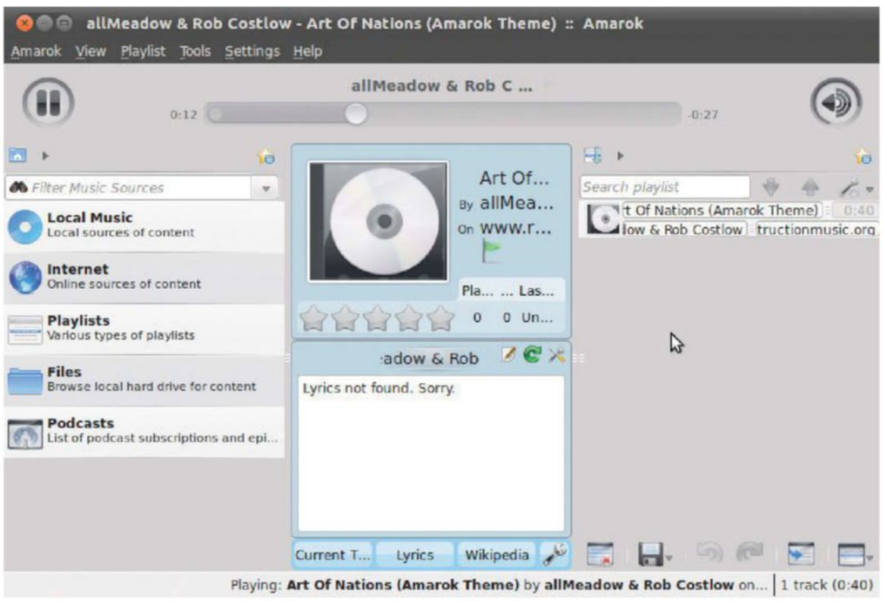
Sadly, there is a big hole in Amarok's functionality, and it's the same one that Rhythmbox has: its music-centric approach means that there's little to no support for video files. Although recent releases of Amarok have added support for playback of music videos, it's very much a basic interface – and the development team have stated several times that they have no intention of turning Amarok into a general media player like Banshee. If you're looking for advanced video playback features or support for DVDs, you'll need to look elsewhere.

There is good news, however: by not attempting to create a jack-of-all-trades, the Amarok team have been able to create an impressively feature-rich music player. As with Rhythmbox and Banshee, Amarok includes built-in browsers for the Jamendo music catablog along with Last.fm support and the ability to browse and play back music held on Ampache music servers.

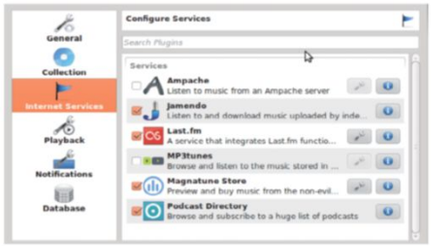
For power users, Amarok includes an impressive scripting language that allows playlists to be generated based on terms like 'tracks from around the year 1998'. An impressive array of tagging and sorting facilities makes library management a breeze, while the PopUp Dropper menu system makes many tasks significantly easier thanks to a drag-and-drop system.

For those running Amarok on a GNOME desktop, there's good news: the built-in notifications system, which uses KDE's Plasma, can be switched off and replaced with native notifications – a must if you're running Amarok on an Ubuntu system, as we were during our test.

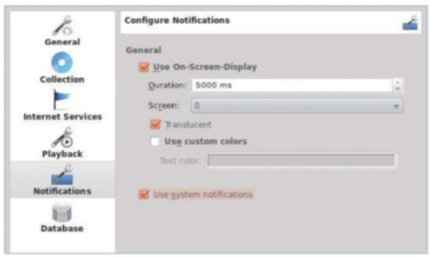
One thing we did notice, however, was performance: compared to the other packages on test, Amarok was one of the most demanding on system resources and by far the slowest to load – even when we switched to a native KDE system. While the performance was still more than acceptable, it's something to note if you're running on older hardware.



The main interface of Amarok is a clean, three-pane view



Amarok includes excellent support for online services like Last.fm and Ampache



The built-in notifications can be disabled in favour of system notifications

DETAILS

Features
Amarok is feature-rich, but the lack of video support means it's not for everyone

8

User interface
The user interface is clean, and the KDE-specific notifications can be disabled for GNOME users

File compatibility
Audio file compatibility is great, but there's very limited video support

7

Overall
For those looking for a native KDE application, Amarok is a great alternative to Rhythmbox

More information
http://amarok.kde.org/

GROUP TEST

Photo management apps

Darktable

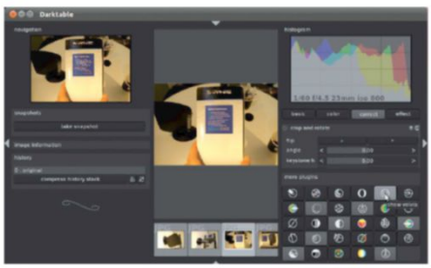
A relatively new package, Darktable's aim is to be a FLOSS alternative to Adobe's Lightroom

Only founded in 2009, the Darktable project is already proving popular among photographers thanks to its impressive feature list and modern user interface.

Installation is straightforward, with binaries available for most distributions – and a PPA for Ubuntu users. Sadly, few distros include Darktable in their default repositories... yet.

Unlike the other applications on test, Darktable doesn't offer any file management functionality – beyond a single button for deleting rejected images entirely. Instead, it relies on you to deal with the organisational aspect, and instead concentrates on what its creators term 'workflow management.'

The main interface is the digital representation of a light-table – the light-up boxes used by film photographers to help organise images. Photos can be imported into the 'film roll' – although they're never actually moved anywhere on the disk, but merely included in a database file that tracks their locations – and viewed in an impressively smooth, zoomable interface.



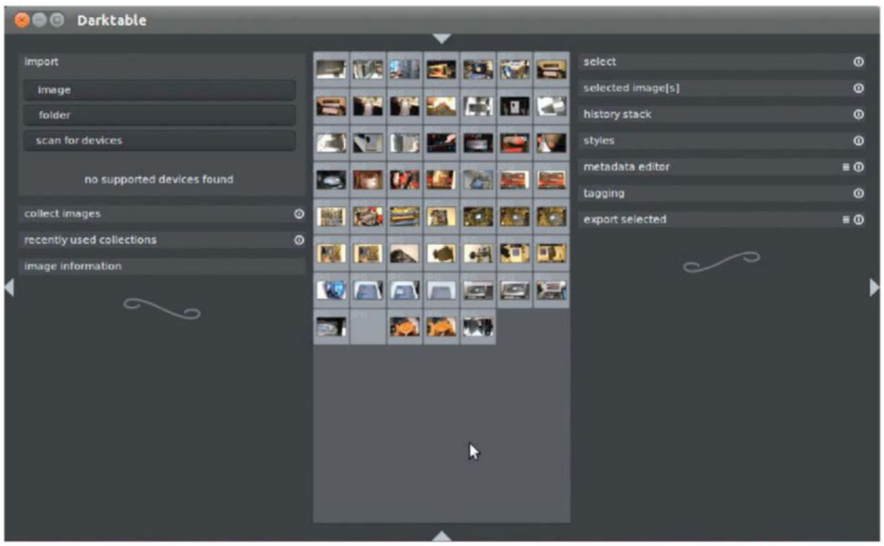
Darktable's editing features are the most comprehensive of any package on test

Although Darktable can be used to manage low-dynamic range formats such as JPEG, it comes into its own when used with the RAW format images saved by most DSLR cameras. The image editing and enhancement capabilities are second to none, and provide a digital darkroom for the 'development' of each image – and the loading of RAW files is very quick.

Speed is one of Darktable's best features, in fact: while it requires a reasonably powerful machine for the editing of higher-resolution images, it includes GPU acceleration as standard – and when browsing through the light-table mode, you can really feel the difference.

Sadly, it's not without its flaws. The package locked up when we told it to create a high-dynamic range image from a single JPEG – a daft request, to be fair, but not one that should have required deleting the database file to resolve.

Despite this, the sheer volume of features available in Darktable make it easy to recommend – and for those who do a lot of work with RAW format images or who are looking for something as close to Adobe's popular Lightroom as possible, it's particularly appealing.



The main user interface in Darktable emulates, ironically enough, a light-table

DETAILS

Installation
While Darktable hasn't made it into many distributions repositories yet, binaries are available – along with an Ubuntu PPA

User interface
With a clean UI, Darktable is a real looker, but benefits from use on higher-resolution displays

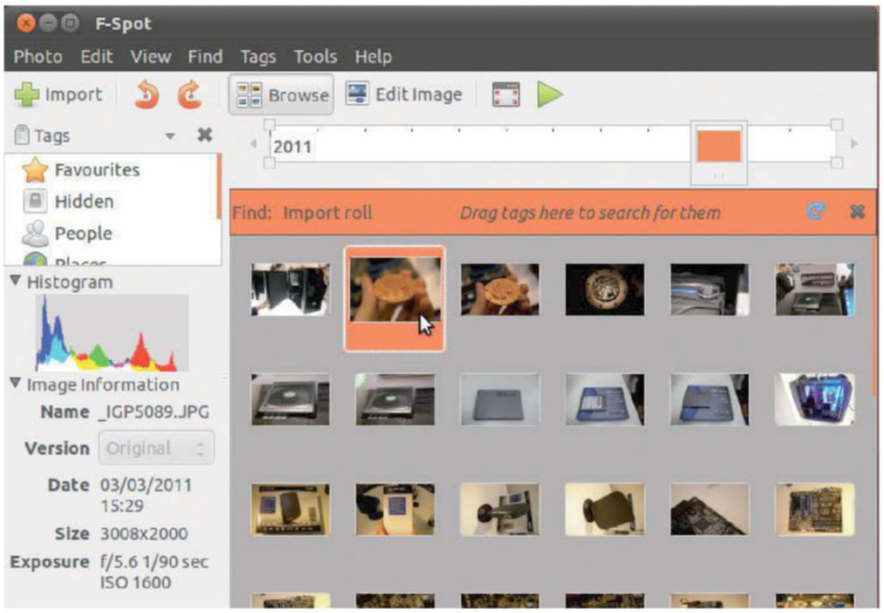
Photo editing features
A massively comprehensive selection of editing tools – especially for RAW images

Overall
Great for RAW users looking for a Lightroom equivalent, but poor for file management

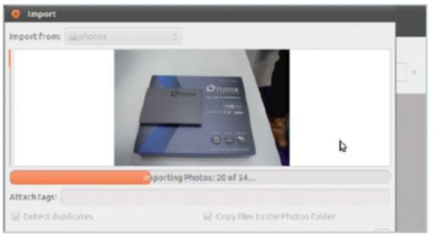
More information
http://darktable.sourceforge.net

F-Spot

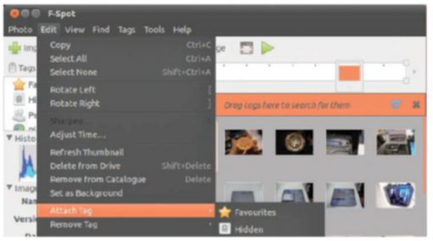
The default photo manager for Ubuntu until recently, F-Spot still has a lot to recommend it



F-Spot's main user interface is clean, though feels a little dated



F-Spot's import process is fast and includes de-duplication facilities



F-Spot's tagging features are simple, but offer a good way of organising your photos

F-Spot was, for the longest time, the default photo management application in the popular Ubuntu distribution, before being ousted by Shotwell in more recent releases. While Canonical may have decided that F-Spot isn't the number-one choice for photo management any more, it's still a pretty capable piece of software.

For most users, installation of F-Spot is a breeze, with the package included in many distributions' repositories – but it is heavily GNOME reliant, meaning those running alternative desktops such as KDE will have a harder time of things.

Once installed, F-Spot offers a simple user interface which hides a surprisingly powerful photo management system. During importation – from a camera, external storage device, or a local or remote directory – users have the option to move files automatically into their Photos folder, detect duplicates and parse any subdirectories that may be found. This latter feature is especially welcome for those who have already had a stab at manually organising their image collections.

Once imported into F-Spot's database, the photos are easy to find: a timeline at the top means that it's quick to jump to specific sections, while a powerful tagging system allows images to be assigned events, places and people – or even to be hidden from view altogether.

When you've organised your photos, tagged and labelled them, and you're ready to start tweaking images, F-Spot includes integrated image editing functionality – but, sadly, it's not terribly impressive. Although the usual suspects – red-eye reduction, colour adjustment, sharpness masks, sepia and greyscale modes, and cropping – are all included, none is particularly powerful. It's not F-Spot's aim to replace a fully-featured image editing application, however – and for those who simply want to tweak their images a little, the integrated editing functionality will be more than adequate.

F-Spot is fast, and lightweight enough to run on even more specification-challenged systems – but always remember to be aware of the requirements for GNOME libraries to be loaded if you're using a lightweight desktop on a machine with less RAM than usual.

Sadly, the F-Spot project is starting to feel its age. There's nothing particularly wrong with the package, but its simple user interface and somewhat limited toolkit make it easy to see why Canonical would choose to move elsewhere. However, if you're not expecting the world's most powerful editing tools, it's certainly still worth trying out if you're a GNOME user.

DETAILS

Installation
Simple for GNOME users, but users of other desktop environments may struggle

User interface
8
The user interface is clean and fits in well with a GNOME desktop, but feels a trifle dated

Photo editing features
7
Basic editing features are included, but more advanced users will want to look elsewhere

Overall
F-Spot is lightweight and capable, but is outclassed in terms of features by some of the other packages on test

More information
http://f-spot.org

digiKam

The only KDE-centric app on test, digiKam has some impressive features for amateurs and pros alike

■ digiKam includes good help and a modern user interface

digiKam is an impressively powerful piece of software – and it's immediately after the installation process, which should be straightforward for KDE users and only slightly more difficult for users of other desktops, that the flexibility of the software makes itself obvious via a handy quick-setup wizard.

Once that's complete, it's time to import your photographs – and while the usual suspects of cameras, scanners and directories are supported, digiKam also includes the ability to import from web services including Picasa, Facebook and SmugMug.

■ The setup wizard walks new users through configuring digiKam

The user interface is slick, although to get the most out of it you'll want to integrate it into a KDE 4 environment - however, it still performed admirably on our GNOME testbed system, with no obvious slowdown. Tooltips did prove a slight problem – rendering too dark to be seen – but this can be corrected with some fiddling in the settings.

Where Shotwell and F-Spot let themselves down on the image editing functionality they offer, digiKam truly excels: full metadata editing is supported, there's a dedicated geolocation option for viewing or altering the embedded geotag on supported image formats, and there's also an excellent selection of quality-boosting features – including ones for red-eye reduction, hot pixel correction and noise reduction.

It's true that the actual image editing functions don't extend quite to the level of those offered in Darktable, but they're extremely impressive nevertheless – and combine well with a fully featured file management system, something that its slicker rival sorely lacks.

For those loading from lossy file types such as JPEG, digiKam even includes the facility to attempt to procedurally reduce compression artefacts – making it an excellent tool for those with cheaper cameras or smaller memory cards, or for images imported from websites such as Facebook.

As well as its impressive import functionality, digiKam has possibly the most comprehensive export facility around. Images – or entire albums – can be exported to HTML, Flash, iPods, remote computers, or uploaded to sites including Flickr, Piwigo, Picasa, Facebook, SmugMug and Zoomr.

The only complaint we could really find with digiKam is that it can be, at times, a little overwhelming. With the sheer versatility and flexibility on offer, it can leave new users feeling a little stranded – although its clever layout makes it relatively easy to use only the features you're familiar with, discovering new functions as you progress in learning the intricacies of the package.

DETAILS

Installation
8
digiKam is in most distros' repositories, although non-KDE systems need libraries installing

User interface
digiKam's UI is clean and uncluttered, and integrates well in a KDE 4 system

Photo editing features
7
There's a wealth of power inside digiKam, although Darktable pips it to the post in this category

Overall
It's hard to find fault with digiKam, and it should be the first choice for KDE users needing a photo management application

More information
www.digikam.org

Shotwell

Designed as a next-generation replacement for F-Spot, Shotwell offers a more feature-rich management app

Another GNOME-centric image management application, Shotwell tries to build on its predecessor with some impressive new functionality. While it doesn't always hit the spot, it's still worth a look – especially if you take video as well as photographs.

Installation is simple, with binaries available for most distributions – and if you're using a current release of either Ubuntu or Fedora, you've almost certainly got it installed already.

There's clearly a reason why Shotwell has become the default photo management application for both Ubuntu – replacing F-Spot – and Fedora, and it's easy to see why as soon as the package loads: the user interface feels both more modern and less cluttered than its Ubuntu predecessor.

Importation is fast from any source – and if you're moving from F-Spot, Shotwell will even drag your old images and tags across for you, so nothing is lost in the move. As with other packages on test, images can be imported from a variety of sources, including digital cameras, external storage devices, and local or remote folders.

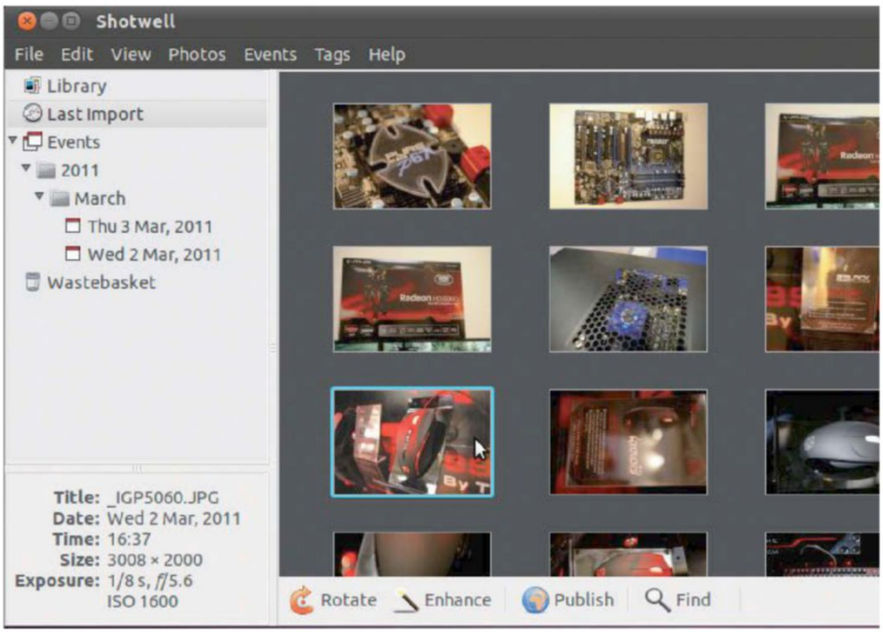
The Shotwell Library splits imported images into 'Events' – and while this is, by default, simply the date on which the photos were taken, custom events can be created to help keep track of which images apply to which real-world event.

A built-in rating system lets you score your images based on quality, and a filter means that you can easily hide the lower-quality shots without actually deleting the images from your hard drive. Individual images can also be titled, again without affecting the original filename.

One of the biggest features Shotwell has over F-Spot is its 'Publish Photos' option: a built-in system for uploading edited images to selected online photo sharing services. While the selection is somewhat limited – with just Facebook, Flickr, Picasa and Piwigo available by default – it's a nice touch, and makes getting your holiday snaps online significantly easier.

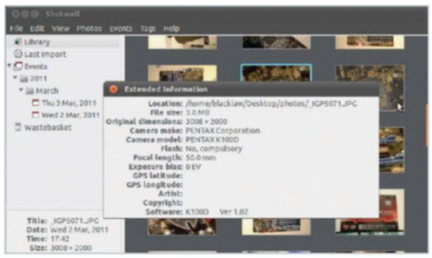
Photo editing, sadly, isn't Shotwell's strong point. A simple 'Enhance' option is a welcome addition – offering users a one-click method to quickly improve under- or over-exposed images – but otherwise options are somewhat limited, even when compared to F-Spot.

That's not to say that Shotwell is of no use to more advanced photographers, though. The package can be linked to an external image editing program, like GIMP, and used simply for image management. Those who want an all-in-one solution, however, should look elsewhere.



■ The user interface in Shotwell is clean and modern

■ Shotwell's import process is quick and easy, and includes file information



■ Extended image information is quickly accessible in Shotwell

DETAILS

Installation
Binaries are available for most distros, although those not using GNOME require extra packages

User interface
The UI is clean and modern, although not as attractive as alternatives such as Darktable

Photo editing features
The editing functions on offer are somewhat limited, but adequate for most users

Overall
A great choice for many, although more advanced photographers will want a separate image-editing application

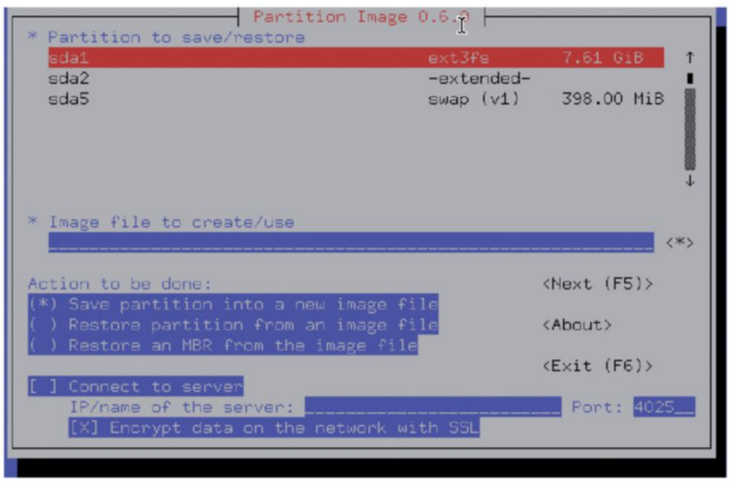
More information
http://yorba.org/shotwell/

GROUP TEST

Recovery live CD supertest

SystemRescueCd 2.0.0

SystemRescueCD is a Gentoo-derived distribution weighing in at 280MB and packing numerous partition recovery tools



Partition Image makes it easy to back up an entire partition or MBR

The Gentoo-based SystemRescueCd is, as the name suggests, a bootable live CD that is specially designed to help rescue a non-booting Linux system.

The most recent version, released on 1 January, features the 2.6.35 kernel, which includes support for systems running pretty much every file system – plus support for network file systems based on Samba, NFS, or over SSH.

In order to maximise the possibility of bringing back a dead system, the CD ships with a variety of boot options – despite its diminutive size at just 280MB. As well as console boot options with both 32-bit and 64-bit kernels, the SystemRescueCd comes with a graphical environment complete with the popular Firefox web browser for looking up additional help on the internet.

For systems that have failed due to errors in the partition table, the SystemRescueCd is a real boon: in addition to the graphical GParted utility, it includes cfdisk, fdisk and GRUB, along with Christophe Grenier's powerful TestDisk scanning package which analyses a storage device for deleted or missing partitions and allows the partition table to be rewritten or lost files restored.

File system recovery tools for most common Linux and Windows file systems are included, and for those having problems with their bootloader, the inclusion of GRUB Legacy, GRUB 2 and LILO is a real boon, covering most eventualities.

With many of the recovery packages included with SystemRescueCd being console based, and the default boot option being to drop to a console rather than a graphical user interface, it's no surprise to see that a selection of text editors are included: as well as the X-based Geany, the disc ships with Vim and the compact Emacs-alike Zile installed as standard.

When the disc is being used to recover data, restored files can be saved to internal storage, external storage, a network share, or burned to a CD. For systems that only have one CD drive, the SystemRescueCD includes an option to load the entire disc into RAM, allowing the live CD to be ejected once booted – but you'll need at least 512MB for that to work.

The SystemRescueCd was quick to boot on our test system and supported every storage device thrown at it. Reinstalling the GRUB 2 bootloader on our 'damaged' install went without a hitch, and the TestDisk package was able to find and restore a partition we'd deleted – despite the age of the program, which is the April 2009 release.

DETAILS

Features

While not being the most feature-rich distribution in our test, for most common Linux issues SystemRescueCd has a package to suit

Ease of use

The GUI mode and included web browser makes things significantly easier, although tools like TestDisk can be confusing at first

Flexibility

Although including some hardware test packages, SystemRescueCd is more suited to disk-related disasters

Overall

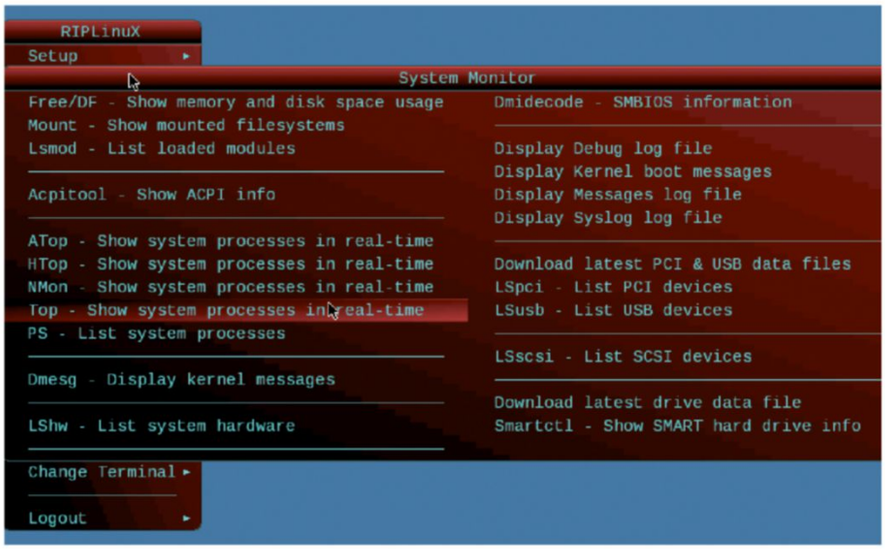
SystemRescueCd is a lightweight Gentoo derivative, quick to boot and suited to both console-jockeys and GUI-lovers alike

More information

www.sysresccd.org

Recovery Is Possible 11.3

At just 125MB, Recovery Is Possible is the smallest distribution on test – ideal if you've only got a small USB memory stick spare



■ RIP's System Monitor menu provides access to many handy analysis tools

If you're on a capped broadband connection, or you've only got a small storage device to play with, Recovery Is Possible could be one of your only options: at just 125MB, it's the smallest disaster recovery distribution in our test group, but don't think that means that it isn't feature-packed.

Shipping with both 32-bit and 64-bit versions of the 2.6.35 kernel, as with SystemRescueCd, most hardware and file systems are again supported natively.

While the menus aren't as immediately clear as those on SystemRescueCd, the on-screen help does a good job of walking you through the options available when you first boot to a console.

Breaking the software options down into categories, the console-based menu system - activated with 'pdmenu' - does an excellent job of directing you where you need to go, and there's a wide selection of tools on offer.

As with the other distributions on test, tools for editing, recovering, and imaging drives and partitions are provided as standard – along with the useful TestDisk package and PhotoRec, which is specifically designed for searching damaged hard drives and memory cards for recoverable photographs and other image files.

While some of the tools accessible from the console-based menu are a little superfluous – such as the calendar, or the option to view who is currently online on the system – it is, in general, well laid out.

Moving into the GUI, the amount of software that is packed into the distribution is unbelievable for its size. As well as the console-based tools, Recovery Is Possible (RIP) includes three graphical web browsers: Firefox, Chrome and Opera.

Unlike other discs on test, RIP doesn't start networking by default – and while it's only a menu-option away in GUI or console modes, it's an additional step that may catch out the unwary.

For the more advanced user, the lack of GRUB Legacy's interactive mode may irk, but most other utilities are in place. Sadly, the small footprint of the distribution does mean some harsh choices have been made – and while Vim and nano are both present and correct, Emacs fans are left without any of the variants of their favourite text editor.

As is to be expected, RIP handled all the hardware we threw at it, and its extremely small size made it quick to burn to a CD or copy to a USB memory stick in an emergency.

■ The console-based pdmenu in RIP is handy for when a system doesn't support X

DETAILS

Features

The small footprint results in some restrictions, but the developers have done an excellent job of cramming in lots of handy packages

Ease of use

The console-based menu is comprehensive, but no default networking and an unclear GUI could leave new users confused

Flexibility

While RIP doesn't include the most comprehensive toolkit around, most common scenarios are catered for

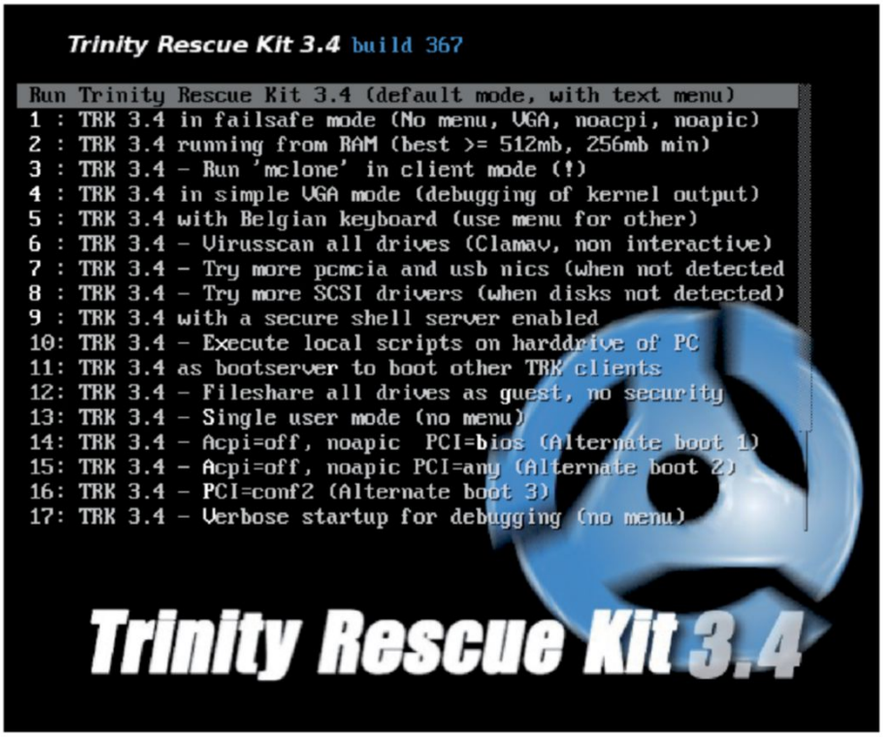
Overall

The RIP team has done an amazing job cramming such a powerful toolkit into such a small size, but in terms of polish and features it can feel a little restricted

More information
http://rip.7bf.de/current/

Trinity Rescue Kit 3.4 build 367

The Trinity Rescue Kit fits remarkable power into a 146MB download, with in-depth documentation for each feature



The Trinity Rescue Kit's boot menu has plenty of choice for systems that struggle to boot

Upon booting the Trinity Rescue Kit, or TRK, the immediate feeling is that it has been developed largely with Windows users in mind – and while many of the tools accessible from its console-based menu, such as virus-scanning and junk file cleanup, are certainly focused on repairing a Windows system that has predictably gone wrong, its Linux basis makes it a good choice for anyone who uses multiple operating systems.

For users of mclone, the disk and partition cloning package, TRK is an excellent tool: from a single menu option, mclone can be executed in either server or client modes to create exact copies of a partition for backup, forensic or recovery purposes. Coupled with the option to load TRK as a network boot server – allowing other systems on the network to boot into TRK without a CD with a PXE-compatible network card – it's an excellent feature.

Again, the use of the 2.6.35 kernel means that most hardware is supported as standard, and while the menu is biased towards recovering a Windows system, the user has the option to ditch the menu and drop to a root shell for Linux system recovery.

Unlike the other recovery distributions on test this month, TRK doesn't ship with a GUI, and while it's understandable given the size of the ISO, it's a shame that it couldn't include a cut-down version.

More seriously, while GRUB Legacy and LILO are both included with TRK, GRUB 2 is nowhere to be seen – meaning that the disk can't be used to recover the bootloader on GRUB 2-based systems without replacing it outright.

The lack of a GUI means that only console-based web browsers are available, which can make searching for additional help and advice a challenge. But in TRK's favour is the extremely comprehensive documentation included on the disc, which – while written from an enthusiast's perspective rather than a professional's, and containing frequent typographical and grammatical errors – covers much of what is possible with the included toolset.

It's hard to recommend TRK to Linux users, as much of the focus is on repairing Windows installations, but as long as you don't need GRUB 2 or a GUI, it's possible to use TRK to repair both operating systems.

Like the other discs in our test, TRK coped admirably with all external and internal storage devices thrown at it and we were able to find and restore our missing partition table using the May release of TestDisk.

DETAILS

Features

While the Windows-specific utilities could come in handy, we'd have traded them for a GUI, GRUB 2 and extra text editors

Ease of use

The menu is very Windows-oriented, and the lack of GUI makes life harder for non-experts

Flexibility

For users who are asked to repair Windows and Linux systems, TRK offers a comprehensive toolset in a small package

Overall

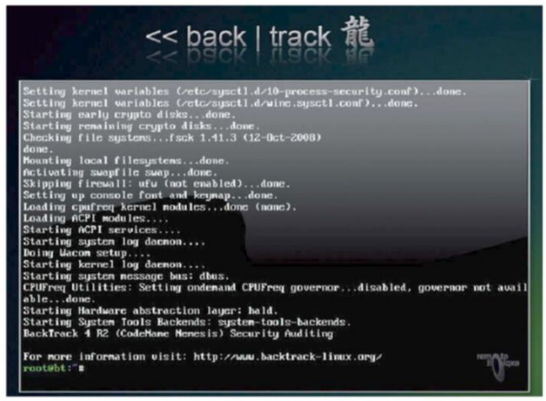
TRK is a pretty polished distribution – despite the help files, which are in sore need of proofreading – but more suited to those who administer both Windows and Linux systems

More information
http://trinityhome.org

BackTrack 4 Release 2

At 1.9GB, BackTrack 4 is by far the largest distribution on test, requiring a DVD or USB stick to run

■ BackTrack's boot screen is visually pleasing, but lacking in options



■ The default Terminal in BackTrack includes a background image, but plainer versions are also available

While BackTrack 4's immense size, at 1.9GB, might seem a trifle excessive, when booted the reason for its seeming bloat becomes clear: this Ubuntu-based distribution packs everything bar the kitchen sink.

While BackTrack has been created from the perspective of a penetration tester, it contains numerous utilities designed for getting data off systems that are seemingly beyond hope, as well as diagnosing issues with various hardware.

Although the boot menu doesn't have the number of options offered by the other recovery discs on test, once booted the power and flexibility of the system becomes clear – and can be a trifle overwhelming.

The developers of BackTrack have worked hard to include as many data recovery and security-related tools as possible, from radio network analysis software for Bluetooth and 802.11 networks and packet capture tools to forensic-grade drive imaging tools and file carving utilities.

Sadly, for system recovery purposes, many of the utilities are superfluous. Although network sniffers, exploit toolkits and backdoor utilities are handy tools to have for a security analyst or penetration tester, they're not much use in a disaster recovery scenario.

Likewise, much of the size of the distribution is due to the developer's decision to include every possible package for each eventuality. While the other distributions on test offer one or two

■ The sheer volume of utilities available in BackTrack can be a little overwhelming

packages for searching for missing partitions or files, BackTrack includes a massive 16 different tools, from the professional-grade Autopsy to the TestDisk and PhotoRec packages we've seen on other recovery distributions.

While this can be seen as a pain – especially if you're trying to download the ISO in a hurry after something's gone wrong – it offers immense flexibility, and allows the user to use whatever package they're most comfortable with rather than forcing them into a predetermined choice.

The good news is that for hardware or disk failure, BackTrack offers a wide range of utilities designed to track down every last bit of recoverable data – and includes the option to install the OS as a permanent single- or multiple-boot option on your system.

During test, BackTrack was by far the slowest of the bunch, struggling with the – admittedly weedy – test hardware and taking an inordinate amount of time to boot. However, once loaded it proved more than capable of recovering our missing data and saving it to external storage.

DETAILS

Features

There's no denying that BackTrack packs a lot of tools into its belt, but it's missing bootloaders and 32-bit/64-bit kernel options

Ease of use

Many of the tools don't work without additional configuration, and the sheer volume of packages is overwhelming

Flexibility

The lack of boot options means that a quick repair often takes longer than it should

Overall

BackTrack is an incredibly powerful distribution, but using it for disaster recovery feels a little like using a Swiss Army knife to crack a nut

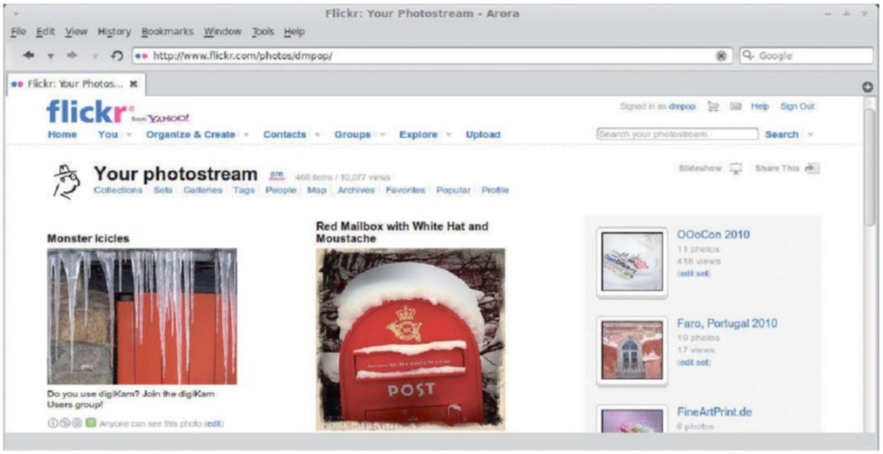
More information
www.backtrack-linux.org

WEB BROWSER GROUP TEST

Battle of the browsers

Arora

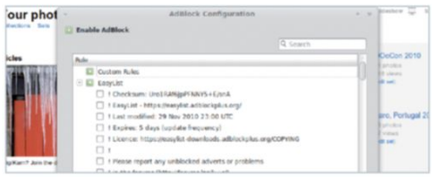
A lightweight WebKit-based browser – what it lacks in features, it makes up for with speed



It's easy to dismiss Arora as some obscure web browser no one has ever heard of. But if you are looking for a browser that is light on resources and not overloaded with features you never use, then Arora is definitely worth a closer look. It is still at an early stage of development, but it's already stable enough for daily use. Arora is built for speed, which you notice as soon as you launch the browser. On our test machine Arora started mind-bogglingly fast, beating all other browsers in the test. It also feels lightning fast in day-to-day use, rendering even complex pages almost instantaneously. While Arora is not the most tweakable browser out there, it sports a resizable Location bar and the movable main toolbar, so you can place it anywhere you like.

Arora shares a lot of similarities with the Midori browser. Arora is based on the same WebKit rendering engine (more precisely, Arora uses the QtWebKit port of WebKit) and offers a similar basic feature set. Like Midori, Arora provides support for multiple search engines, and you can add custom search services of your own. You can also specify keywords for search engines, so you can perform searches from the Location bar. Similar to Midori, Arora provides basic bookmark managing capabilities and it lacks any form of data syncing.

■ Arora renders webpages faster than the other browsers tested



■ The built-in AdBlock lets you blog those annoying adverts

Unlike Midori, though, Arora doesn't support extensions, so you are stuck with the default functionality. But on the bright side, this also means that you can't compromise the browser's speed and stability by installing third-party modules. Although Arora doesn't support extensions, it does come with a built-in ad blocker and the ClickToFlash plug-in. The former does an excellent job of filtering ads while the latter helps you to manage Flash content.

When it comes to privacy, Arora is not so different from Midori either. You can delete your browsing data and enable the private browsing mode. However, Arora offers better cookie control. You can specify which cookies you want to accept, how long the browser should keep the cookies, and set the expiration time for them. In addition to that, the browser allows you to filter tracking cookies. Arora doesn't allow you to change user agent, which can be problematic if you're visiting sites that perform a browser check before letting you in.

Despite its decidedly limited feature set, Arora is a surprisingly capable browser that is perfectly suited for use on older hardware, laptops and netbooks.

DETAILS

Usability
Besides support for tabs and multiple search engines, there are no features that improve your browsing experience

Extensibility
No support for extensions, although Arora does include ad and Flash blockers

Privacy Controls
Besides the private browsing mode and the ability to delete browsing data, Arora boasts decent cookie management features

Stability
For a browser that is still in a very early stage of development, Arora is surprisingly stable

Overall
A lightweight browser with a decidedly limited feature set, but lightning fast

More information
code.google.com/p/arora

Google Chrome

A browser with some clever features, including the Omnibar

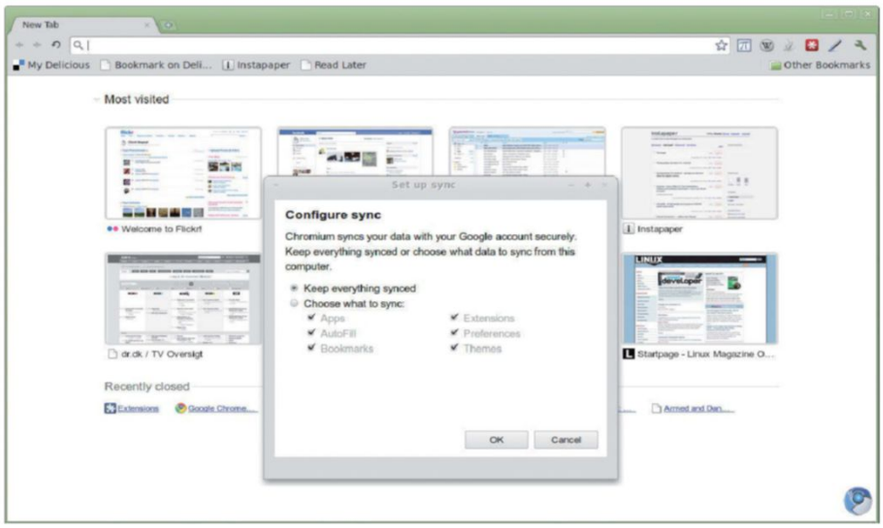
Google Chrome is a relative newcomer on the browser scene, but it has already established itself as a viable alternative to its more mature competitors. Several factors contributed to Chrome's meteoric rise, and Google's marketing muscle is only one of them. In many ways, Google Chrome feels like a browser done right. It sports an unobtrusive interface, it feels fast and it offers a few creature comforts that make browsing more efficient. For example, Google Chrome was the first browser which had a built-in feature for synchronising bookmarks and preferences between multiple browser installations. Besides bookmarks and preferences, the current implementation of the synchronisation feature lets you sync extensions, themes and form data. You need to have a Gmail account for the synchronisation feature to work, though.

Omnibar is another clever feature that makes life easier. It acts both as an address bar and a search field. When you type a search term in the Omnibar, it returns search results from your browsing history and from the Google search engine. The Omnibar has another clever trick up its sleeve. Type the address of the website you want to visit and if it has its own search feature, you can use it right from the Omnibar. For example, type 'amazon.com', hit the Tab key, enter the search term you want and press Enter. The browser will take you to the Amazon page containing the search results. In addition to that, the Omnibar recognises certain keywords. Type, for example, 'maps' followed by an address and Chrome will use the Google Maps service to display the specified location. Other keywords include calendar, gmail, tube and pics.

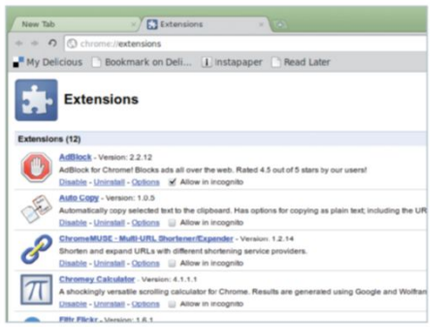
Each tab in the Chrome browser runs as a separate process, which offers several benefits. The key among them is improved stability, as a misbehaved tab doesn't take the entire browser down. Chrome sports its own task manager tool which lets you monitor and manage all running tabs. There is even a 'Stats for nerds' link that generates a detailed overview of all tabs and their memory consumption. Speaking of tabs, you can 'pin' any opened tab, which reduces its size and keeps it open permanently.

Chrome also makes it ridiculously easy to turn any webpage into a standalone app. Click on the wrench icon, choose the Tools>Create Application Shortcuts command, press Create and Chrome will create a desktop shortcut or menu entry which launches the specified page in a separate minimal browser window.

Of course, as any self-respecting browser, Google Chrome supports extensions, and you can install and remove extensions without restarting the browser. While the collection of Chrome extensions is still relatively modest compared with what's available for Mozilla Firefox users, it's only a matter of time before Chrome catches up with Firefox, both in terms of numbers and quality.



Google Chrome lets you sync not only bookmarks and form data, but also extensions and themes. The synchronisation feature works perfectly, but it does require a Gmail account



Google Chrome supports extensions, and you can install and remove them without restarting the browser

DETAILS

Usability
It sports dozens of big and small genuinely useful features which significantly improve the browsing experience

Extensibility
Chrome supports extensions, and there are hundreds of modules to choose from. The browser makes it easy to install, manage and remove extensions

Privacy Controls
The browser lets you wipe off your browsing data and it sports the Incognito mode

Stability
Chrome is rock solid, even when loaded with extensions

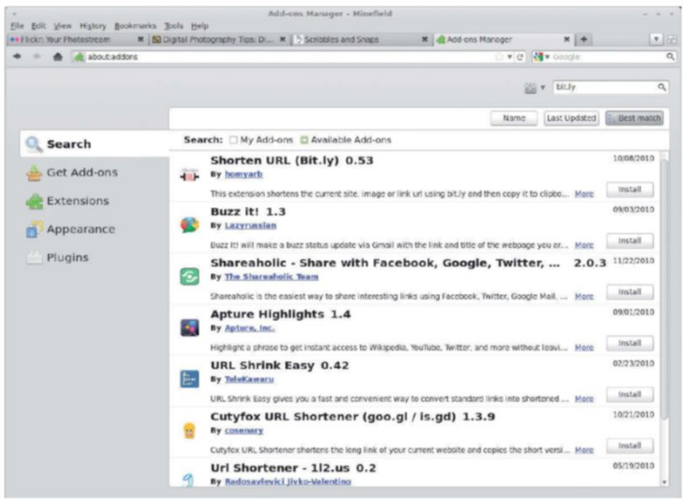
Overall
A supremely stable and fast browser with some innovative features

More information
www.google.com/chrome

Mozilla Firefox

The latest version of the popular browser has a few new tricks up its sleeve

■ The brand-new Tab Groups feature makes it supremely easy to organise and manage multiple tabs. You can't save the tab groups, though



■ Firefox boasts a redesigned Add-ons Manager which you can use to manage existing add-ons and install new ones, as well as browse and search the add-on repository

DETAILS

Usability
A lot of usability improvements, but a few shortcomings too

Extensibility
The redesigned Add-on Manager makes it easy to extend the browser's default functionality by installing add-ons

Privacy Controls
All the key features to protect your privacy are present

Stability
It's still a beta version, so crashes do occur

Overall
The latest Firefox is much improved, though still not perfect

More information
www.mozilla.com

The Firefox browser needs no introduction. It's the default browser on many mainstream Linux distributions and it's the most popular browser, bar Internet Explorer, on the market. The upcoming 4.0 release has been in the making for quite a while, but it's already stable enough to be included in our group test instead of the current 3.x series version.

Besides a refreshed and tweaked interface, Firefox 4 sports a few features designed to improve the browsing experience. The most prominent addition here is the Tab Groups feature which allows you to pool and manage multiple tabs. This is, indeed, a handy tool, but it can't save tab groups you've painstakingly created and arranged. If Tab Groups could do that, it would have been a killer feature. Similar to Chrome, the Location bar in Firefox 4 can be used to perform searches using the default search engine. But Firefox provides the dedicated search bar, too, which seems like a waste of space to us (you can remove the search bar manually, though).

Firefox's default functionality can be extended using add-ons, and the official add-on repository contains thousands of nifty modules for you to try. The latest version of the browser sports a completely redesigned Add-ons Manager that simplifies the process of installing and managing add-ons. You can now search, browse and install add-ons right from within the Add-ons Manager – a great time saver in our book. However, the biggest annoyance is still there: once you've installed an extension, you must restart the browser to activate it. Mozilla does provide a new Jetpack SDK for building add-ons that don't require restart, but these add-ons are still far and few between.

Firefox 4 adds synchronisation capabilities, so you can now sync bookmarks, passwords, history, and tabs across several machines. The only omission here is the lack of support for syncing add-ons. The good news, though, is that your data is encrypted, and you are not tied to a Gmail account or a specific service as you are in Chrome.

Although privacy controls in Firefox 4 haven't been tweaked too much, the browser's privacy functionality still remains strong. You have full control of your browsing data and cookies, there is a private browsing mode, and the browser encrypts sync data.

These are just a few interesting features in Firefox 4 that are exposed to end users. But the browser sports a slew of under-the-hood improvements as well, including support for HTML5 and the WebM video standard, better font rendering, a faster JavaScript engine and much more. All in all, the upcoming release has enough new features and improvements to keep Firefox in the game.

Midori

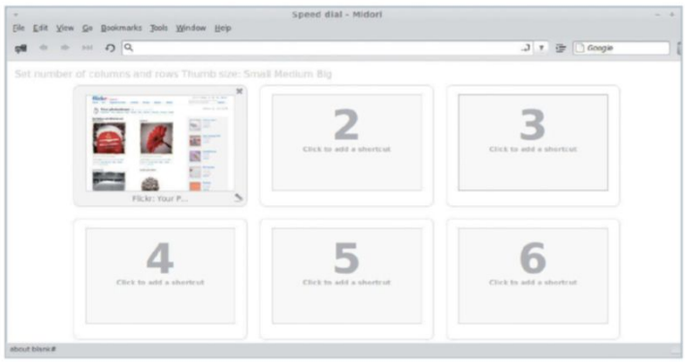
A basic but functional WebKit browser with a GTK-based native look

The name Midori may not ring a bell with many Linux users, but this browser is rapidly becoming a popular choice with lightweight Linux distros thanks to its speed and modest system requirements. While Midori lacks the advanced features of mainstream browsers, all the essential functionality is there. It is based on the WebKit rendering engine and provides full integration with the GTK toolkit. This means that the browser sports a native look on any Linux distro using a GTK-based graphical environment.

Midori sports a speed dial feature which lets you add shortcuts to often-used websites. These shortcuts are displayed when you open a new tab. While the browser has a separate search bar, you can also use the main address bar to perform searches. Type a search term in the address bar and the browser conveniently displays a list of search engines to choose from. Alternatively, you can use keywords to perform a search using the search engine you want. For example, enter 'g monkey' if you want to search for the word monkey in Google, or 'wp monkey' to look up the word in Wikipedia. In addition to the default search engines, you can add your own search services and specify keywords for them.

Midori offers basic bookmarking functionality, but it has no synchronisation capabilities whatsoever, so you're better off using services like Delicious to keep your bookmarks in sync. The browser does support extensions, but you are limited to the modules bundled with it. The supplied collection of extensions is pretty limited, but it includes a few must-have tools like an ad blocker, a feed reader and a user add-ons extension. The latter allows you to extend the browser's functionality by installing user scripts.

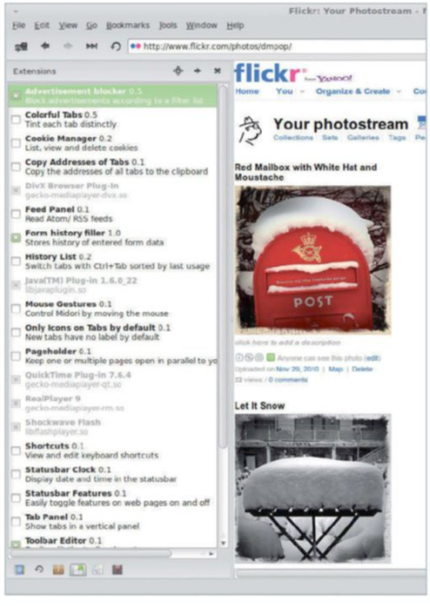
■ The Midori browser can handle multiple search engines. In addition to the default engines, you can add other services and specify keywords for them



■ The Speed Dial feature allows you to add shortcuts to often-used Web sites. This way, you are presented with links to your favourite places every time you open a new browser window or tab

Similar to other browsers in this test, Midori allows you to delete private data, and you can do that either manually or configure the browser to clear the data automatically. There is also an option to automatically remove cookies older than the specified time limit. The latest version of the browser also provides support for private browsing. However, a private browser window looks exactly like a regular window, which makes it somewhat difficult to identify it. Midori also makes it easy to change the user agent, so that websites and services can identify the browser as Firefox, Internet Explorer or Safari. You can even specify your own custom user agent string.

While Midori does a commendable job of rendering webpages, it has some stability issues. During our test, the browser crashed several times for no apparent reason and without any explanation. On the plus side, the browser prompts you to restore or reset the previous session after the crash. To be fair, the browser is still at an early stage of development, so we don't hold occasional crashes against it.



■ Midori ships with a handful of extensions that can add new features to the browser. With the user add-ons module enabled, you can extend Midori's functionality by installing user scripts

DETAILS

Usability
Midori includes a few creature comforts like the speed dial and support for tabs

Extensibility
With Midori, you are limited to the extensions that are bundled with the browser

Privacy Controls
The browser lets you remove cookies and private data. It also sports the private browsing mode

Stability
The browser crashes occasionally for no apparent reason

Overall
Has its plus points, but not the most stable nor feature-packed browser

More information
www.twotoasts.de

VIRTUALISATION GROUP TEST

Which hypervisor is right for you?

Choosing a hypervisor based on price or popularity is a bad idea. Virtualisation is a multidimensional technology, the choice of which shouldn't be limited to such esoteric characteristics...

KVM: The Kernel Virtual Machine

Red Hat's branded virtualisation method that's built into the Linux kernel

Red Hat's purchase of Qumranet in 2008 brought it firmly into the world of virtualisation. A full virtualisation solution, KVM can host almost any operating system that can use standard PC hardware as a medium. Tested guest OSs include Windows, Linux, Solaris/OpenSolaris, Free/Open/NetBSD, Minix, QNX, Plan 9 and DOS.

Since kernel release 2.6.20, KVM has been included as part of the base Linux kernel. However, KVM isn't a simple installation, nor is it something a new sysadmin should attempt. It involves working at the command line, installing several prerequisite packages (see below) and possibly patching the kernel. **KVM's list of prerequisites and requirements are:**

1. A 64-bit CPU (multi-processor, multi-core recommended) with virtualisation-capable extensions (Intel VT or AMD SVM).
2. Linux kernel 2.6.20 or higher that includes the KVM modules.
3. Kvm-kmod for older kernels or those without KVM.
4. Zlib libraries and headers.
5. SDL libraries and headers.
6. Alsa libraries and headers.
7. Gnutls libraries and headers.
8. Kernel headers (kernel-devel).
9. QEMU (qemu-kvm).

Once installed, an administrator may work at the command line or install a separate Virtual Machine Manager (VMM) application – this graphical interface has no frills, but delivers a more pleasant user experience than CLI.

■ Preparing for installation of CentOS 5 in a KVM virtual machine

One of KVM's outstanding features is its extreme performance – due, in part, to the SPICE (Simple Protocol for Independent Computing Environments) protocol. The use of SPICE as a local and remote display environment, Red Hat believes, will enable enterprises to move to a virtualised desktop environment.

SPICE isn't the only reason for KVM's extreme performance. Red Hat's Enterprise Linux (RHEL) performance as a guest on its native KVM touts an amazing 92 to 140 per cent of native (bare metal). These numbers include heavy workloads such as Oracle, Microsoft Exchange Server, SAP, LAMP and Java.

KVM is free by design but fee-based commercial support from Red Hat is available. Businesses with mission-critical applications and services need to engage Red Hat for premium support options.

KVM is an enterprise-level virtualisation solution that fits well into any size organisation and for any application.

DETAILS

Installation and support
Installation requires some reading and advanced command-line skills

Ease of use
Use the Virtual Machine Manager and not the command line if ease of use is important

Virtual machine management
The Virtual Machine Manager is all business and no frills

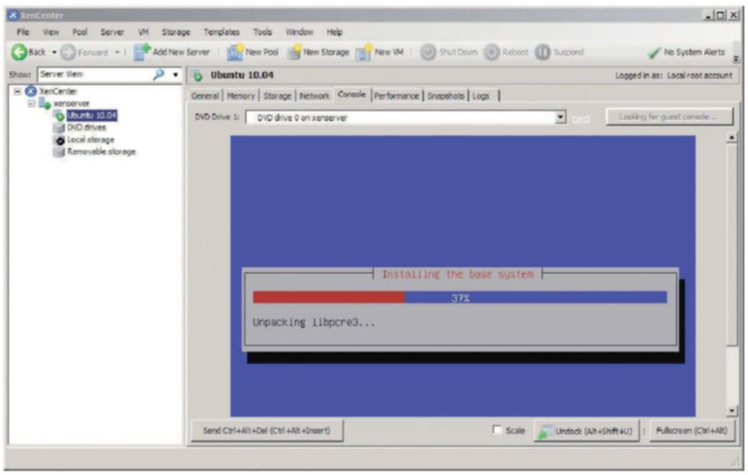
Features and capabilities
KVM offers incredible performance for operating systems and applications

Overall
High performance and low cost makes KVM a good choice for companies of any size

More information
www.linux-kvm.org

XenServer

A free product with high aspirations and excellent technical support



Installing a new Ubuntu 10.04 Server system virtual machine using XenCenter

After Citrix Systems acquired XenSource, in 2007, Citrix was quick to integrate the Xen product and philosophy into its current offerings. Since that time, Xen-related virtualisation products have stolen the show from the old Citrix product line-up and now almost everything it produces bears the Xen moniker. Server virtualisation has perhaps passed Citrix's application virtualisation as a sustainable product line. Citrix is also heavily marketing its XenDesktop product as a virtual desktop solution for enterprises, making its 'desktop as a service' idea a reality.

XenServer is a free download from the Citrix website. It takes longer to download the ISO and burn it to disc than it does to install a XenServer host. Once downloaded and burned, you boot the CD image on a qualified host system and tap the keyboard a few times to complete the installation. Within minutes of boot, an enterprise-capable virtualisation platform awaits its destiny as a virtual machine host. The XenServer console isn't terribly useful for anything but looking at system statistics, rebooting, shutting down, configuring the network or setting up SSH access to the host. Virtual machines can't be created or managed from the XenServer console.

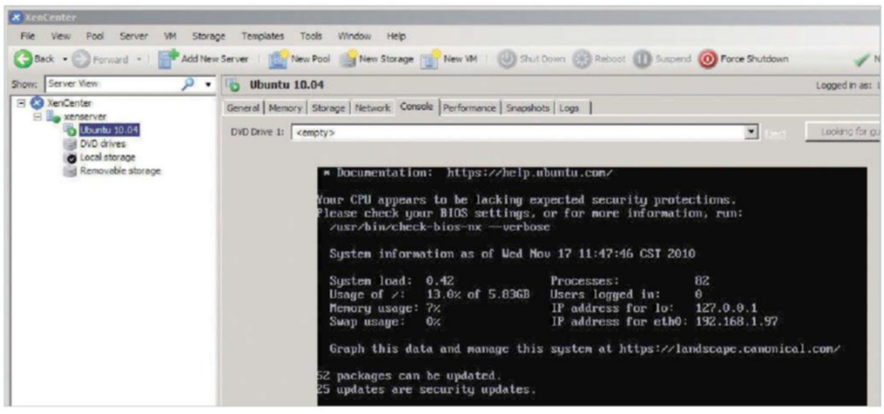
To manage a XenServer environment, you need to download and install the XenCenter application. XenCenter only runs on the Windows platform, so at least one Windows system must exist on the network in order to act as a management station.

XenCenter installs easily onto any contemporary Windows operating system (Server or Workstation). From within this application, almost every aspect of a virtual machine is configurable, including console access, performance and storage connectivity. Note that there are some features unavailable in the free version that can only be unlocked with a purchased licence. XenCenter is easy to use, most of its actions are wizard-driven, and the online help is actually helpful.

Command-line management is a completely different story than its user-friendly XenCenter graphical cousin. Command-line interaction is cryptic and difficult to use except for the most mundane tasks. Stay clear of the command line unless administrators have had extensive training, possess advanced skills or have their résumés up to date.

One of XenServer's most compelling features is that it supports a wide range of hardware – great for those without huge hardware budgets. It runs well on commodity-priced systems.

For those on tight budgets, XenServer is an excellent path to realising a virtual infrastructure dream.



Working at the Ubuntu 10.04 Server virtual machine's console inside XenCenter

DETAILS

Installation and support
The most difficult part of this installation is accepting how easy it is to install

Ease of use
The XenCenter interface is friendly, but the command line isn't

Virtual machine management
Management is easy, but the useful parts only appear after purchasing a licence

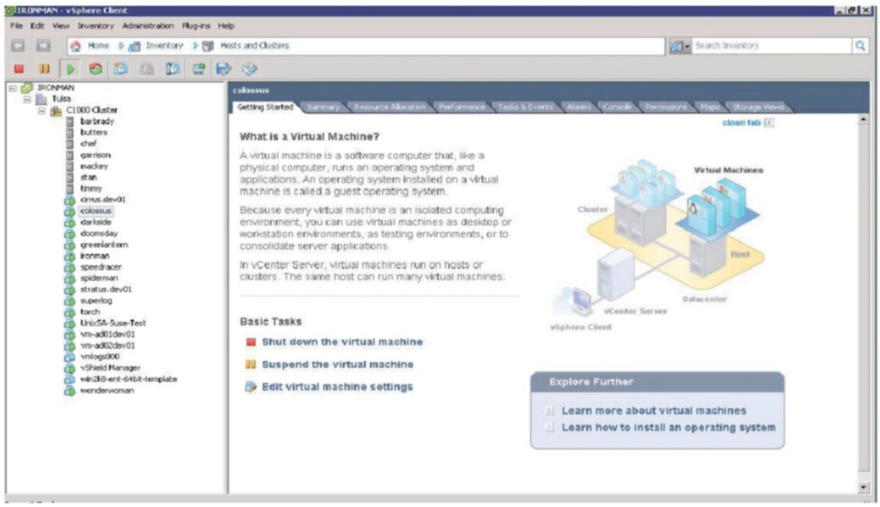
Features and capabilities
Don't expect higher-end features from the free product

Overall
XenServer is a good product and can be used free of charge; but if requirements demand, it will cost a bit of cash

More information
www.citrix.com/xenserver

VMware vSphere

Packed with features, it isn't as easy to use as some, and it's aimed at larger organisations



VMware vSphere Virtual Center and full virtual machine inventory

VMware has held the number one spot in the world of server virtualisation for several years. It's no wonder with its almost ubiquitous ESX/vSphere product, overwhelming third-party vendor support and major partnership with Cisco. VMware is the frontrunner in the virtualisation space – a situation that isn't likely to change in the foreseeable future. That said, it does have its problems. No product is perfect and VMware's vSphere is no exception.

Installation is very easy for those familiar with Linux installations and still quite simple for those who aren't. The only part of installation that delivers any amount of angst is the bit to do with datastores (storage). For those acquainted with virtualisation terminology (datastore, LUN, remote storage, local storage), installation generally proceeds without issue.

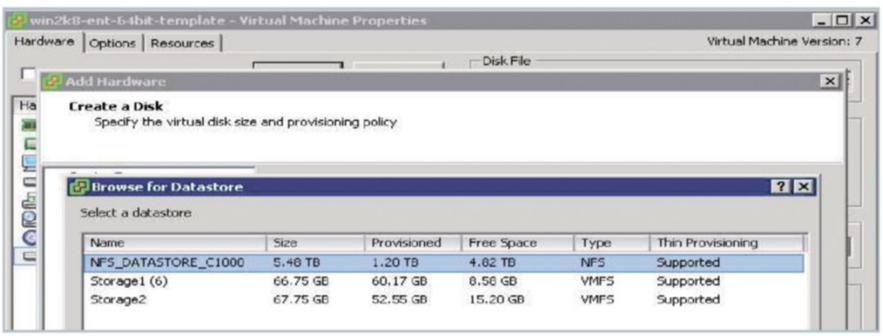
VMware offers training for those interested in pursuing certification or in-depth knowledge of installation and virtual machine management the VMware way. This training is highly recommended for those struggling to gain on-the-job experience. Production VMware environments are no place for neophytes.

VMware environments can become quite complex with data centres, datastores, virtual switches, virtual host clustering and dynamic resource scheduling (DRS).

Management of virtual machines takes place via the graphical vSphere Client or in a Windows PowerShell command line. Most administrators use the Windows-only vSphere Client and are possibly unaware that the VMware host is running Linux. Once an administrator has gained competence with VMware's methods, the vSphere Client is efficient for VM management, although working at a VM console through the graphical interface can be quite sluggish.

The standout VMware feature is the ability to map LUNs (SAN storage units) to virtual machines. This feature allows administrators to select and use storage in raw or VMFS format, which is advantageous for those who wish to virtualise certain database workloads.

VMware offers top-notch support, stable products and a huge user base. Although it is a costly alternative to other types of virtualisation, that fact hasn't slowed its adoption or expansion in the world's data centres. Its third-party vendor support sets it apart as the leader in this space, and its popularity certainly isn't a downside. VMware's server virtualisation products are suitable for larger data centres and firms whose server numbers run into the hundreds. It isn't appropriate financially for small server environments of 50 or fewer systems.



DETAILS

Installation and support
Installing vSphere is as easy as any Red Hat Linux installation

Ease of use
VMware's vSphere requires training to attain competency

Virtual machine management
Centralised management makes vSphere a tough competitor

Features and capabilities
VMware's products contain every imaginable feature and capability

Overall
VMware's vSphere is an outstanding product, but it isn't built for small firms

More information
www.VMware.com/vSphere

OpenVZ

Linux containerised virtualisation delivers native performance

OpenVZ is container (or operating system-level) virtualisation. This type of virtualisation was popularised by Solaris in its use of Zones. The commercial firm Parallels supports the OpenVZ project and uses OpenVZ technology as the basis of its Parallels Virtuozzo Containers product. A container is a chrooted directory structure that shares compute resources (CPU and memory) with the host system, but uses its own separate set of files and applications to isolate it from the host. A container can be rebooted, have its own users (including root), have its own IP addresses, processes, files and libraries. A container can also be independently patched.

Container virtualisation enjoys the highest performance of any type of virtualisation because all applications run as native on the system using the single system kernel. Since this type of virtualisation relies on a single, shared kernel, a host system can only host guests that can run the Linux kernel.

OpenVZ installation is non-trivial and requires advanced sysadmin skills and root access to the host system. It involves replacing the Linux kernel with an OpenVZ-enabled one, rebooting into the new kernel, disabling SELinux, tweaking kernel parameters, installation of control utilities (vzctl) and quota management software (vzquota).

VM management from the command line with OpenVZ is also complex. Although powerful, the command-line OpenVZ utilities are somewhat difficult for those new to the terminology and concept of container virtualisation. A better method is to install WebVZ, which is a web-based OpenVZ management utility. The installation of WebVZ is non-trivial as well as it requires Ruby, Ruby Gems, Ruby on Rails and the WebVZ package installed via Ruby.

■ Creating a new OpenVZ container virtual machine using WebVZ

Once installed and configured, OpenVZ offers a powerful, efficient and inexpensive virtualisation method that leverages existing hardware better than any other virtualisation technology. VM densities can run into the hundreds for standard commodity-based hosts.

Virtual machines can be created from ISO images, from bootable CD/DVDs attached to the host or remote systems, or from downloadable operating system templates.

One big downside to this type of virtualisation is that each container is host-bound. That is to say, containers aren't able to migrate from one host to another in case of required host reboots. If the host is rebooted, all containers are also rebooted. A potential remedy to this is to create all containers on shared storage, such as SAN, between two or more clustered systems. This scenario would guarantee zero downtime for all of the hosted containers.

■ Browsing OpenVZ container virtual machines in the WebVZ Interface

DETAILS

Installation and support
Installation requires advanced setup and reading from the OpenVZ website

Ease of use
Avoid command line difficulty with WebVZ – a web-based OpenVZ interface

Cloud & social features
Experience excellent management via the command line or WebVZ interface

Software
OpenVZ is a proven enterprise solution for Linux virtual machines

Overall
Superb performance, lightweight installation, rapid deployment and complete control over every aspect of virtual machine creation puts OpenVZ in a class by itself

More information
http://wiki.openvz.org

About the mag

The magazine for the GNU generation

Written for you

Linux User is the only magazine dedicated to advanced users, developers & IT professionals

In-depth guides & features

Written by grass-roots developers & industry experts

4.5GB DVD every issue

Four of the hottest distros feature every month - insert the disc, reboot your PC & test them all!

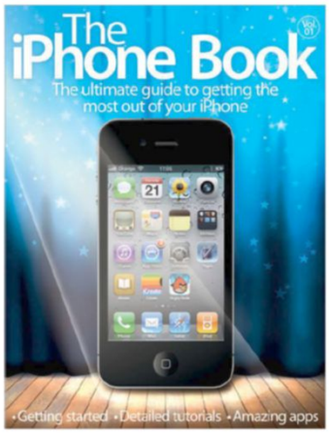
subscribers to...

LinuxUser & Developer™

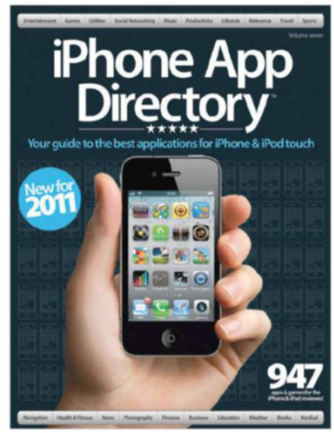
Try three issues for £1 in the UK*
or just $9.72 per issue in the USA**
(saving 37% off the newsstand price)

For amazing offers please visit
www.imaginesubs.co.uk/lud
Quote code ZGGZIN

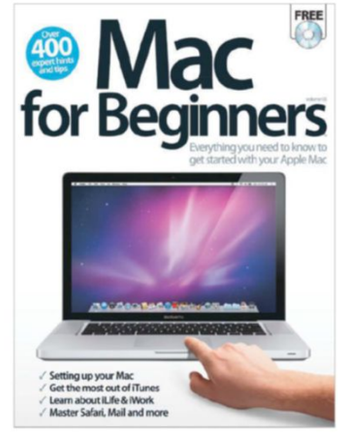
Or telephone UK 0844 249 0282 overseas +44 (0) 1795 418 661



The iPhone Book vol 1
Whether you are new to the iPhone or have had one for a while, **The iPhone Book** is the ultimate resource for getting the very best from your Apple device.
SRP: £9.99



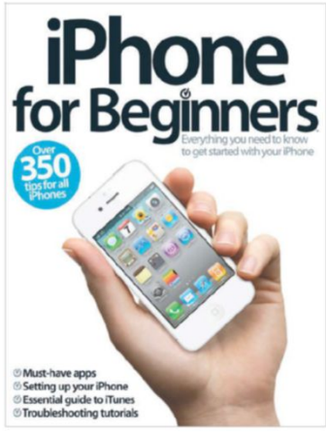
iPhone App Directory vol 7
The latest collection of iPhone apps are reviewed right here, including the very best available for iPhone 4.0, with every single App Store category featured inside.
SRP: £9.99



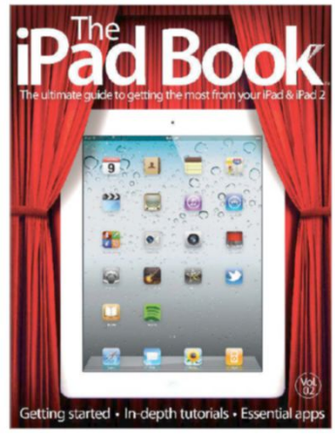
Mac for Beginners 2011
Starting with the basics, this essential guide will teach you how to master all aspects of switching to Mac including OS X, Snow Leopard, Mail and Safari.
SRP: £12.99

Retro Gamer Collection vol 5
An unmissable selection of in-depth articles featuring timeless games and hardware. From Zelda to Asteroids, this book covers all the classic games from days gone by.
SRP: £9.99

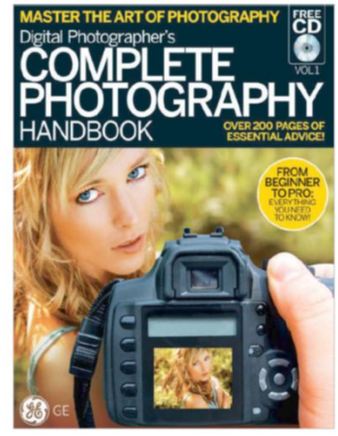
The world's best cre to collect and keep



iPhone for Beginners
Everything you need to get started on your iPhone. With step-by-step tutorials, the 100 essential apps and a troubleshooting guide, this is a must-have for iPhone owners .
SRP: £9.99



The iPad Book
The ultimate guide to iPad and iPad 2, this comprehensive book brings you a wealth of productivity, entertainment and lifestyle tips, along with all the top apps to download.
SRP: £9.99

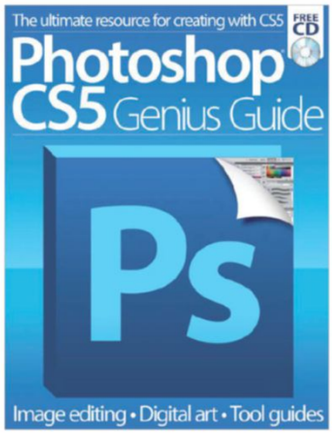


Complete Photography Handbook vol 1
With fantastic shooting ideas and a wide variety of practical tips, this tome is the only resource for digital photographers.
SRP: £12.99

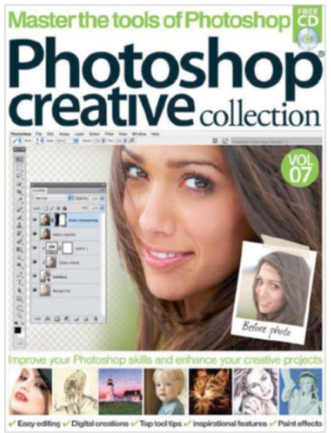
iPhone Tips, Tricks, Apps & Hacks vol 4
Step-by-step tutorials and in-depth features covering the secrets of the iPhone and the ultimate jailbreaking guide make this a must-own book.
SRP: £9.99

Prices may vary, stocks are limited and shipping prices vary

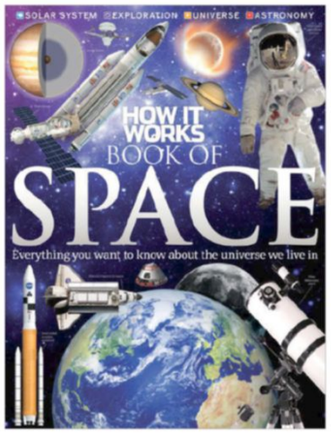
Order online www.im



Photoshop CS5 Genius Guide
Over 200 pages packed full of advice for Photoshop CS5 users, including tool guides, step-by-step workshops and a free CD with over 60 minutes of top video tutorials.
SRP: £12.99



Photoshop Creative Collection vol 7
This latest addition to the Photoshop Collection includes excellent guides to improve your image editing skills whether you're a beginner or expert.
SRP: £14.99

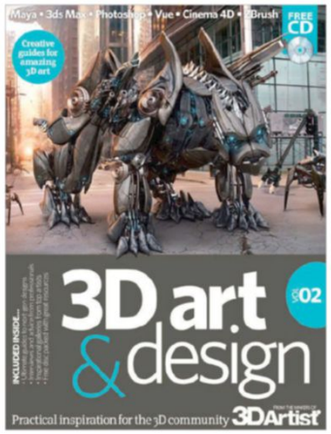


How It Works Book of Space
Feed your mind and fuel your imagination with this amazing guide to space and the universe from **How It Works** magazine.
SRP: £9.99

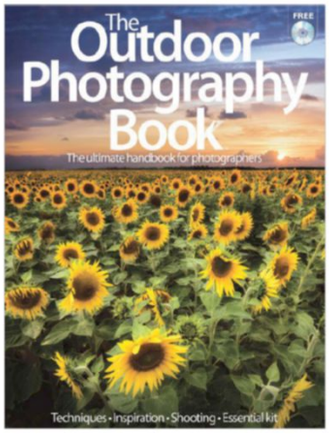
How It Works Book of Amazing Technology
Taking a look into the workings of some of the hottest gadgets and everyday appliances, this is the perfect guide for all tech-lovers.
SRP: £9.99

ative bookazines or give as a gift

IMAGINE PUBLISHING

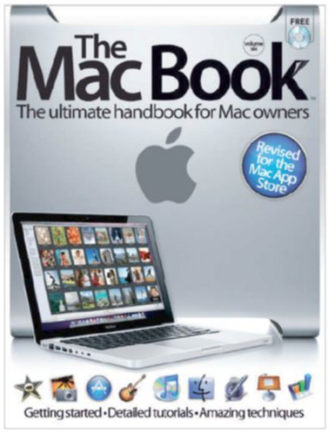


3D Art & Design vol 2
Covering characters, environments, architecture and transport, there is something for everyone interested in 3D design in this book. Become a 3D master in no time at all...
SRP: £14.99



The Outdoor Photography Book
Enhance your photography techniques and learn a range of brand new skills with this essential guide to outdoor shooting.
SRP: £12.99

Web Design 6
The ultimate guide to creative cutting-edge web design. Discover how to produce next-generation web design projects using some of the hottest technologies...
SRP: £14.99



The Mac Book vol 6 Revised
256 pages of practical and creative tutorials and in-depth features that will take you through all areas of OS X, iLife, iWork, Mac App Store and third-party applications.
SRP: £12.99

depending on destination

aginebookshop.co.uk


ASSEMBLING THE AVENGERS
Android Tips, Tricks, Apps & Hacks
iPad App Directory
LinuxUser & Developer
BUILD A FASTER WEB SERVER
Backup & restore
Baking for Beginners
web designer
WORDPRESS SECURITY
iCreate
BECOME A MAC GENIUS
iLIFE UPGRADED
10 ITUNES SECRETS
Android magazine
GALAXY S III
Photography for beginners
25 top tips for better photos NOW!
Advanced Photoshop
CS6
Digital Photographer
SHOOT YOUR BEST-EVER PORTRAITS
X360
BLACK OPS 2
SPEC OPS THE LINE
Apps magazine
THE GREATEST SMARTPHONE APPS
Explore
Read
Shop
SAMSUNG
greatdigitalmags - The official download website of Imagine Publish
HOME
SUBSCRIPTIONS
CONTACT US
Now enjoy yo
mag howeve
Magazines
eBooks
Apps
XBOX SPECIAL EDITION ISSUE
360 MAGAZINE
BLACK OPS 2
GEARS OF WAR 4
EXCLUSIVE ACCESS
3DArtist
ULTIMATE GUIDE TO
27
Tweet
1
✓iPad ✓iPhone ✓Android phone ✓Android tablet ✓Apple Mac ✓Windows PC

No Disc. No Problem

Many of the files you're looking for can be found on the magazine's website

Imagine digital editions are a new and exciting way to experience our world-leading magazines and bookazines.

To get the most out of your digital editions, be sure to enjoy all of our fantastic features, including:

- **Zoomable text and pictures**
- **In-app browsing**
- **Searchable text**
- **Take your collection with you**
- **Read offline**

o buy more Imagine digital editions and for the latest issues and best offers, please go to www.GreatDigitalMags.com

Linux for Professionals

The complete guide to using and developing for Linux, written by respected experts in the open source scene

Linux Masterclass

> Gain essential open source skills

In-depth tutorials

> Step-by-step open source guides

Pro Tips & Tricks

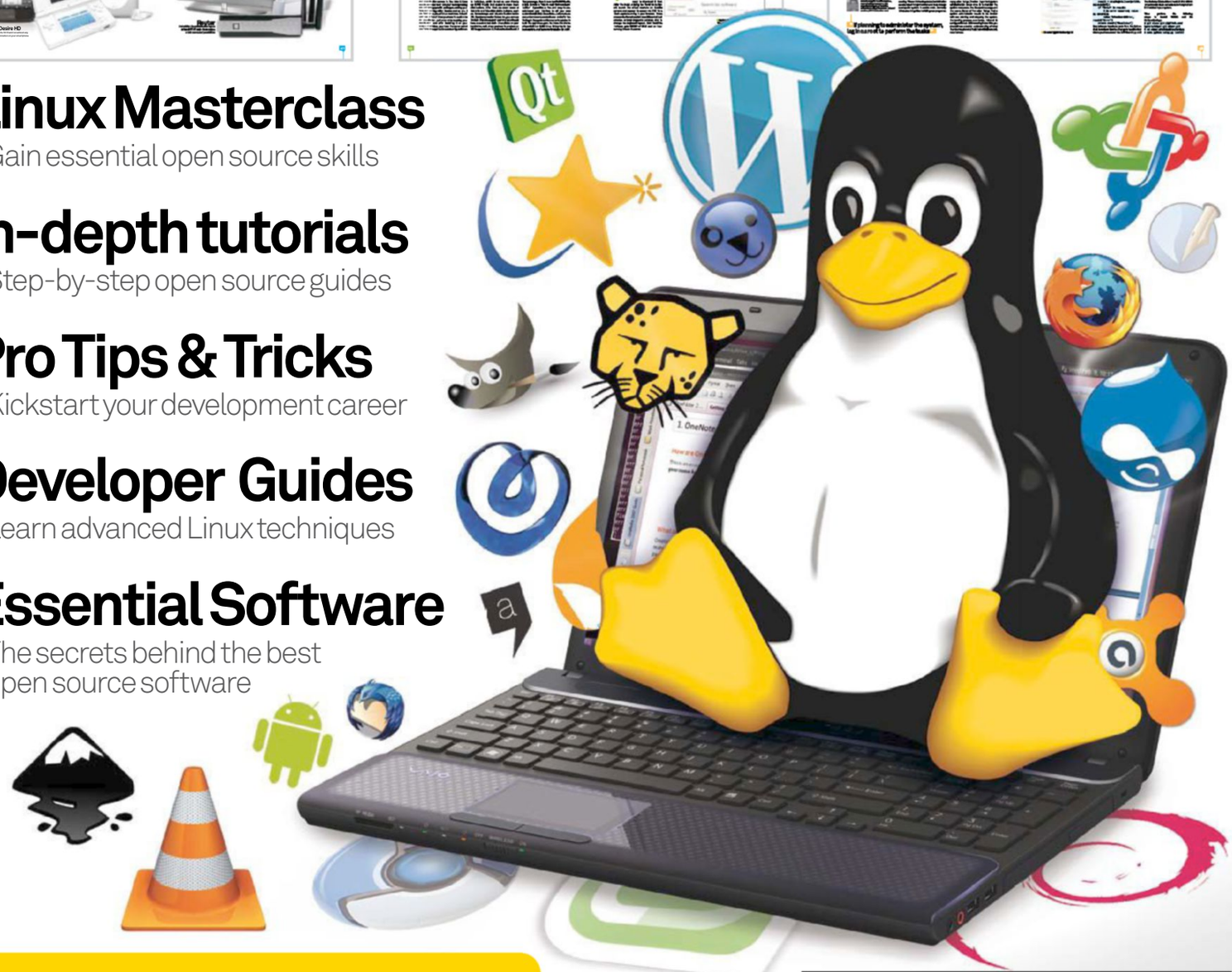
> Kickstart your development career

Developer Guides

> Learn advanced Linux techniques

Essential Software

> The secrets behind the best open source software



4 essential Linux distros

Ubuntu 11.04 · Fedora 15 · Linux Mint 11 · openSUSE 11.4

VOLUME 02 £14.99

ISBN 978-1-908222-2-37

IMAGINE PUBLISHING

9 781908 222237

www.imaginebookshop.co.uk