LXF55 Linux Format July 2004 • LEARN TO LOVE CODE • KOPETE • O(1) SCHEDULER • NEVERWINTER NIGHTS • WEB BROWSER ROUNDUP • MAYA 6 • LDAP • KDEVELOP •

■ HOTTEST NEWS ■ LATEST REVIEWS ■ BEST TUTORIALS ■ YOUR PROBLEMS SOLVED

LINUX FORMAT

THE UK'S BEST-SELLING LINUX MAGAZINE!

Free Linux Pro • Free Linux Pro • Free Linux Pro • Free Linux Pro • Free Linux Pro • Free Linux Pro •

TURN OVER FOR YOUR FREE LINUX PRO MAGAZINE

Learn to ♥ code

Get more from software – master the basics of coding in minutes

» Shell scripting » OpenOffice macros » GIMP filters » KDE » and more!

useful commands • tips and tricks • shell scripting • unix shells • exit codes • file renaming •
adding more swap • programming with ...
writing a script • unix shells • exit codes • ...
Edit View Project Build
Namespace
nvmj.kdevelop
src
nvmj

LDAP BASICS

Network directories and how to use them: complete guide p86

NEVERWINTER ★ NIGHTS! ★

Best fantasy RPG on Linux p18

BETTER BROWSING

Eight web browsers tested to extremes – which is best? p34

"We beat Microsoft to where it was going"

Jeremy Hogan on Red Hat's return to the desktop p06

MAYA 6 ON TEST p26

INSTANT MESSAGING

Kopete talks to AOL ICQ & Yahoo! p64

YOU CAN WRITE A KDE APP!

Easy-to-follow coding tutorial p74

UNRAVELING THE O(1) SCHEDULER

Kernel 2.6's scheduler explained – and why you will care p58

CONNECT TO MS EXCHANGE SERVERS WITH XIMIAN p08

DVD issue also available
Printed in the UK
LXF55 JULY 2004
£5.99
future publishing
Media with passion
9 771470 423019 07
(£5.99 outside UK and Republic of Ireland)

www.linuxformat.co.uk

You can do it!

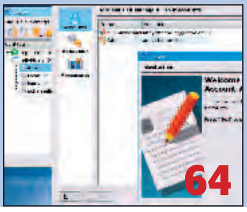
For some reason, a lot of people are afraid of the word 'programming'. Perhaps they associate it with unhappy experiences relating to video recorders.

While it is certainly true that to be a good programmer you need to be able to think in a certain way, almost everybody can manage some level of programming – and the good news is that even the smallest amount can be of great benefit to you. If you can manage to microwave a baked potato, you should be able to create a useful *OpenOffice.org* macro!

The aim of this month's cover feature is to show you just how easy it is to re-write your computer experience with the help of a few scripts or lines of code. From simple *bash* scripts that can automate many tasks through to interprocess communication with KDE, there should be something useful here for everyone. And that's not even counting the regular programming tutorials in the magazine, into which category we could easily put our *GIMP* series this issue, as it continues to discuss custom filter effects with *Script-Fu*.

We are very proud of the fact that this is first and foremost a practical magazine. The ever-changing Linux universe is full of new software to consider and techniques and skills to learn. We certainly learn plenty of things ourselves in the course of researching and writing a lot of the content, and I hope you do too.

You will also note that a new graphic has appeared on this page, indicating our corporate patronage of the Free Software Foundation. Hopefully you will be aware through the magazine of the work of this organisation in promoting the use of Free Software. Obviously, these are goals that we share and we're very proud to be the first magazine to be listed as corporate patrons, and we would certainly encourage those of you who use Free Software (which is all of you) to consider supporting the FSF through donations of time, money or simply by buying things from its store – visit **www.fsf.org** to find out more.



MEET SOME OF *LINUX FORMAT'S* TEAM OF WRITERS...

Andrew Channelle
Newsman and newbies best friend, he shows us how to chat with everyone, be they ICQ, AOL, Yahoo!, MSN...

David Coulson
Our *Answers* guy is a networking and security guru with plenty of sysadmin experience.

Mike Saunders
Journeys right to the edge of the Internet for new Open Source software each month to bring you *Hot Picks*.

Jono Bacon
Core KDE developer, web developer, sound engineer, freelance writer, musician – and very, very tired!

Paul Hudson
He thinks he's a great gamer because he wins the odd game of *Crack Attack*, but time will prove his undoing...

Marco Fioretti
Can't wait to see his work translated back into his native tongue – using *OpenOffice.org* macros, of course!

Hoyt Duff
Leading Linux book author and bringer of righteous indignation to those who don't stick rigidly to standards.

Andy Hudson
Our Advertising guy also writes for the mag 'cause he loves Linux – especially the GNOME way of doing things.

Michael J Hammel
Professional *GIMP* artist who pens (or pencils) our current Open Source graphics tour-de-force.

Jon Kent
Teaches us about the more obscure parts of Linux you've never heard of – like the O(1) Scheduler, for instance.

Nick Veitch EDITOR

AIMS OF THE MAGAZINE

***Linux Format* is a magazine dedicated to Linux and the Open Source community. We aim:**

- To provide the most accurate, unbiased and up to date information on all things Linux.
- To promote the use of Linux in business and the home, for servers and on the desktop.
- To support the Open Source community by providing a resource of information, and a forum for debate.
- To help all readers get more from their Linux experience by providing insightful and useful tutorials.

Corporate Patron
Free Software Foundation

CONTACT US

Letters for publication:
lxf.letters@futurenet.co.uk

Subscriptions/back issues:
subs@futurenet.co.uk

Technical help/Ask the Experts:
lxf.answers@futurenet.co.uk

Disc problems:
lxf.support@futurenet.co.uk

General enquiries:
linuxformat@futurenet.co.uk

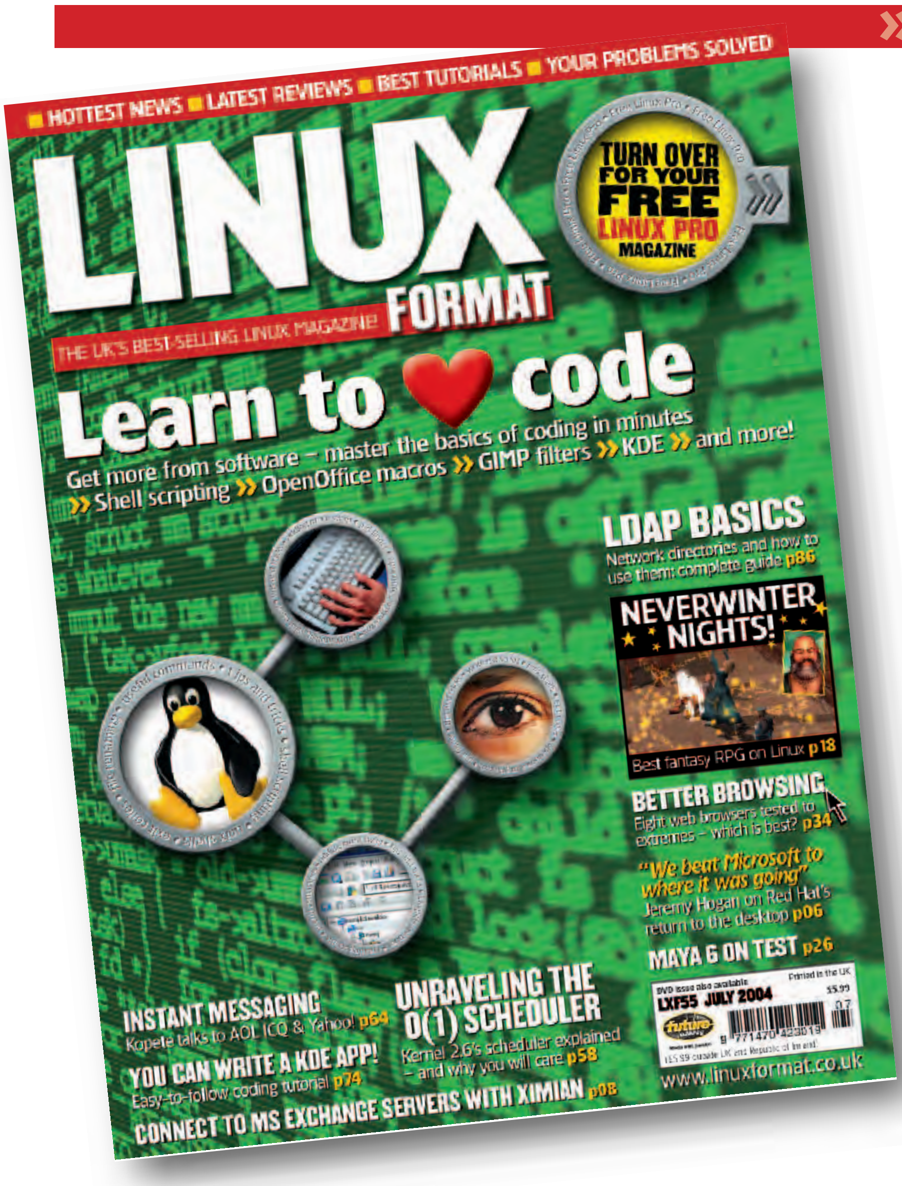
Website: **www.linuxformat.co.uk**
Or send your letters to:
***LINUX FORMAT*, Future Publishing,**
30 Monmouth Street, Bath, BA1 2BW
Phone: 01225 442244
Fax: 01225 732295

More contact info on p95, p99 & p114

Contents

Welcome to another jam-packed issue of *Linux Format*, your guide to all things Linux!

LINUX FORMAT
LXF55 July 2004



HOTTEST NEWS ■ LATEST REVIEWS ■ BEST TUTORIALS ■ YOUR PROBLEMS SOLVED

LINUX FORMAT

THE UK'S BEST-SELLING LINUX MAGAZINE!

TURN OVER FOR YOUR FREE LINUX PRO MAGAZINE

Learn to ♥ code

Get more from software – master the basics of coding in minutes » Shell scripting » OpenOffice macros » GIMP filters » KDE » and more!

LDAP BASICS Network directories and how to use them: complete guide p86

NEVERWINTER NIGHTS! Best fantasy RPG on Linux p18

BETTER BROWSING Eight web browsers tested to extremes – which is best? p34

"We beat Microsoft to where it was going" Jeremy Hogan on Red Hat's return to the desktop p06

MAYA 6 ON TEST p26

INSTANT MESSAGING Kopete talks to AOL ICQ & Yahoo! p64

UNRAVELING THE O(1) SCHEDULER Kernel 2.6's scheduler explained – and why you will care p58

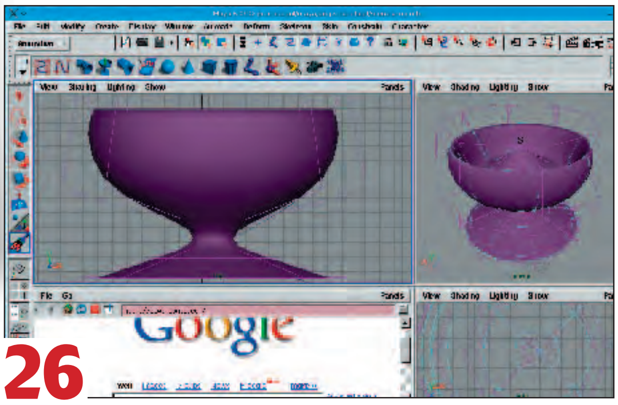
YOU CAN WRITE A KDE APP! Easy-to-follow coding tutorial p74

CONNECT TO MS EXCHANGE SERVERS WITH XIMIAN p08

LXF55 JULY 2004

www.linuxformat.co.uk

» COVER HIGHLIGHTS

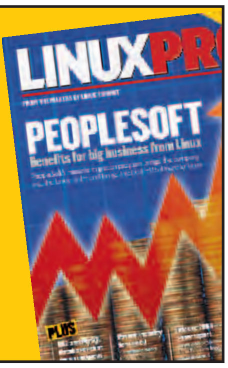


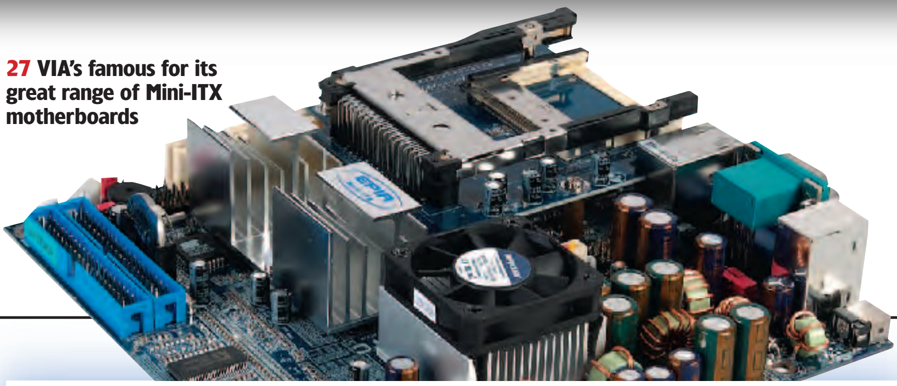
26

'LUGs are great social, technical and fun groups to be a part of – User Groups, *please* send in your announcements...'

'I just don't like being bullied into it and what is – in effect – being lied to. Is this what FOSS is about? *I think not!*'

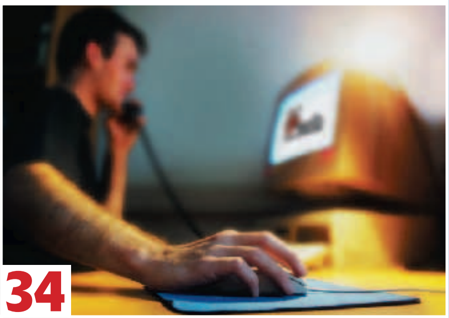
LINUXPRO



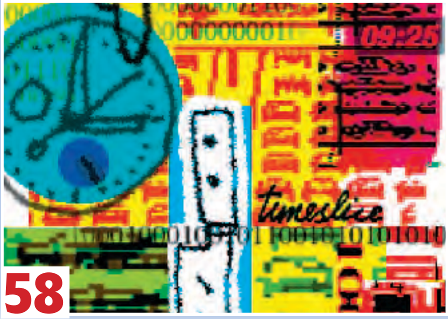


27 VIA's famous for its great range of Mini-ITX motherboards

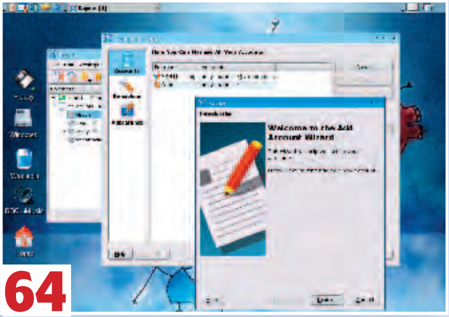
18



34



58



64

Coverdiscs

A DVD or 3 CDs packed full of the latest Linux goodies **105**

CDS A, B AND C

DYNE:BOLIC GNU/LINUX Full-featured powerful multimedia complete distribution
COOPERATIVE LINUX Run Linux apps on Windows *without needing a distro install!*
KDEVELOP All the apps you need to follow our development tutorial series
LEYLINES Read the review – then see whether our extreme opinion is justified!
WEB BROWSERS ROUNDUP including **MOZILLA 1.7, LINKS, EPIPHANY, W3M, KONQUEROR, LYNX, GALEON FIREFOX**
HOT PICKS All the apps from the feature

DVD

OPENOFFICE.ORG 1.1.1 Boost your productivity with all the latest tweaks
XORG-X11 and **XFREE86** Avoid all the licensing furore by switching to the fork
KDETV Watch the football on your PC when everyone thinks you're working!
DSPAM An alternative way to kill UBEs

Please read the coverdisc instructions on page 111 carefully before installing from coverdiscs!

On the CD and DVD

Newsdesk

■ Using MS Exchange with Linux ■ Shrek 2 ■ Lindows goes public ■ Scribus included by SUSE ■ robin ■ Patent nonsense ■ SCO News ■ Wikipedia ■ David vs Goliath ■ Bargain Linux supercomputing

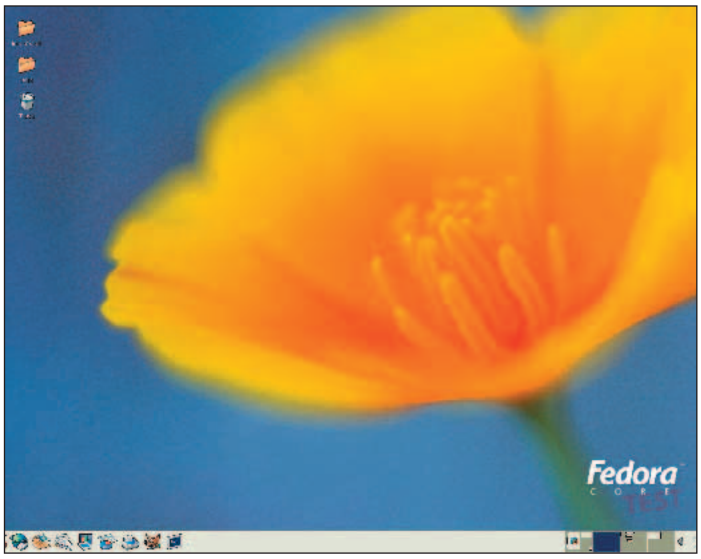
Red Hat's desktop déjà vu

The *"Microsoft of the Linux world"* returns to the commercial desktop market

Red Hat 9 was the company's last crack at the desktop market.

Red Hat executives have announced the company's return to the 'desktop space' – just as its previous desktop product, Red Hat 9, reaches the end of its officially supported life. Speculation has been mounting that the new initiative is in response to Sun's Java Desktop, and the acquisition of SUSE by Novell. Both of these events have the potential to drive Linux adoption in the 'enterprise', the reasoning goes, and Red Hat's previously non-committal stance on the corporate desktop leaves the company in danger of losing its premier status in the US market.

Back in November, Red Hat CEO and President Matthew Szulik suggested that the facilities and applications available on the desktop were not yet ready for prime time, and that home users especially would be better off sticking to Windows. *"I would say that for the consumer market place, Windows probably continues to be the right product line. I would argue that from the device-driver standpoint and perhaps some of the other traditional functionality, for that classic consumer purchaser, it is my view that the technology needs to mature a little bit more,"* he said. Despite the headlines this interview inspired, Szulik also said at the time that he thought *"the enterprise desktop market place is much more strategic and has buyers whose needs we can exceed."*



The Fedora project will be continuing to support 'old' editions of Red Hat desktops through its fedoralegacy site.

Amid accusations that the company had abandoned formerly loyal desktop purchasers to concentrate on money-spinning Enterprise Server products, Red Hat's Community Relations Manager, Jeremy Hogan emphasised the belief that public companies couldn't live on good will alone.

"Many folks think Red Hat goes where the money goes," he said. *"Well, as a public company we have to find money, but the reality is, we beat Microsoft to where it was going. Redmond wants the UNIX space."* This concentration, Hogan says, is not something Red Hat has been given proper credit for by the community.

Hogan added that Red Hat Desktop version 3 (there's a silent 'Corporate' before Desktop, apparently) is the first phase in the company's new strategy. *"We are focusing on things like useability, interoperability with MS servers, directory access, legacy document support, security, hardware and peripheral support for future releases."* These developments would be seen first in the Fedora project before being adopted into the commercial product.

Red Hat Desktop will be available in two flavours: Starter and Extension. The former (split into Proxy and Satellite server versions) is designed for deployments of between 10 and 50 seats and includes the Premium Edition of Red Hat Enterprise Linux AS, while the latter can be purchased in multiples of 50 licenses and is suitable for use on networks which already have a Proxy or Network server. Pricing starts at around US$2,500 for the Proxy Starter edition. The distribution will include all the usual applications including *Evolution, Mozilla* and *OpenOffice.org*, and will also boast fully functional 'out of the box' access to *Real, Flash* and *Citrix* technology.

Official support for Red Hat 9, the company's last desktop release, has now officially ended. Support duties will fall to the Fedora Legacy project – and also to such interested third-party companies as Progeny and Ximian – which will continue to offer security updates to Red Hat customers. Most existing users have been urged to move over to Fedora or – if the job requires it – to Red Hat's Enterprise line of products.

Despite being an 'unofficial' source, the Fedora Legacy project has support from Red Hat and has put in place security measures to ensure that updates come from trusted sources.

www.fedoralegacy.org

Windows Media: Coming to a distribution near you?

At last! A properly licensed DVD player under Linux!

Turbolinux has become the first distribution vendor to ink deals to include both Windows Media technology and a licensed DVD player in a desktop release 10F. Turbolinux is the biggest selling distribution in the Asia-Pacific region, and company President Koichi Yano said the licensing deal will make Linux an even better buy for consumers.

"Japanese consumers are moving in large numbers to Linux," he said. "*Turbolinux Desktop has long had the best Asian language support and with 10F we now have the best multimedia system for consumers."*

Turbolinux 10 has been one of Japan's best-selling software products in 2004, and the company hopes the addition of consumer-centric facilities such as the ability to watch DVD movies and streaming Windows Media content in 10F will improve sales in the home user market.

The DVD system is based on CyberLink's PowerDVD for Linux and is compatible with the industry standard Content Scrambling System (CSS), while support for the WM format has been integrated into the Turbo Media Player, a new product built on the *Xine* multimedia engine.

Turbolinux has also recently signed a deal with Hewlett-Packard to preinstall its '10 Desktop' distribution on new HP Business PCs.

Some pundits opine that the 'F' should stand for a Linux DVD player that's *"Free as in beer"*!

Turbolinux 10F is the first Linux distro to include a 'legitimate' DVD player package, and support for the Windows Media format will be popular, too.

newsbytes

■ **Mandrake** has launched a new service aimed at SMEs, academic institutions and non-profit organisations to provide simple Open Source solutions. Essentially this is a new volume-licensing program with solutions tightly personalised for the purchaser.

■ **Microsoft** will allegedly take the unprecedented step of allowing *"non-paying users of Windows XP"* (*ie* pirates) to update their systems with Service Pack 2 (the security-conscious errata release) in order to prevent continuing problems with worms, viruses and spam relays.

■ Daniel Robbins has resigned his role as Chief Architect of the **Gentoo project**. Sven Vermeulen said Robbins had been the backbone of the project driving development forward at a terrific pace. *"Gentoo's been your child; you gave it birth and watched it grow. It learned from your experience and guidance,"* he wrote on the project's mailing list. *"This and all your contributions have been very well appreciated, by both users and developers."*

■ **The Lawrence Livermore National Laboratory** has recently installed what, on initial testing, may be the second most powerful computer on the planet. Thunder, which cost 'millions of dollars', is built on 1,024 four processor Itanium 2 servers and is rated at 19.94 trillion operations per second, putting it well within the top ten of supercomputers. However, despite the astronomical figures, Thunder is said to have cost about one third of the price of a traditional supercomputer.

■ While **Sun** continues to hum and haw about the possibility of open-sourcing Java, a new debate has broken out about the potential for releasing Solaris under the GPL. Talking about the difficulties of Solaris competing with Linux on the ascendancy, new CEO Jonathan Schwartz said *"Maybe we'll GPL it. We're still looking at that..."* Sun has also been smearing the name of its erstwhile partner Red Hat. During a debate on the possibility of a GPL Java, Schwartz said his problem with the GPL was that it encouraged 'forking'. As an example he mentioned the 'proprietary extensions' Red Hat has made to its Linux AS line. *"Red Hat has pretty much forked the distribution. This has given Red Hat tremendous gains for now, but ultimately it's an impediment in the growth of Linux,"* he said. *"Open standards, which Sun has always supported, are better. Proprietary open source [like RHEL] can come back and bite you."* Baffled Red Hat spokesman Leigh Day said Red Hat Linux was still released under the GPL. *"We are fully committed to Open Source and our code reflects that. Red Hat has no proprietary software built in our distribution,"* he told eWeek.

David Cartwright

An IT consultant since the phrase was respectable, David specialises in Linux systems and solutions.

COMMENT

Decisive desktop

“By the time you read this, Red Hat will have launched its new Red Hat Desktop product. I guess it's about time it did a proper commercial offering for the desktop, although having been told just a few months ago by one of Red Hat's marketing people that the company was concentrating on the server market for the moment, I was a little surprised to hear the announcement of a new desktop released so soon.

The thing is, though, I can't help feeling that this product doesn't really fit into the Red Hat product range properly. There's a comparison chart on the website (www.redhat.com/software/rhel/comparison/) that shows how the various incarnations of Red Hat Enterprise Linux , and one has to wonder what the Desktop flavour brings that the WS ('workstation') edition doesn't – except reduced capabilities such as being limited to a single processor (WS can supports two processors).

I hope for Red Hat's sake the Desktop edition turns out to be a useful addition to the range, and that it's not just there because when it turned off support for good old Red Hat 9, the world went: *"Hey, we were using that!"*. Particularly when you bear in mind that, in my experience anyway, the WS edition is an absolute pig to get working with esoteric hardware when you compare it with the old free version.

There's certainly a place for a non-Windows RH product on the desktop. I do hope, though, that it turns out to be a worthwhile and well-designed offering, not just a half-baked knee-jerk reaction to what Red Hat perceives the market demand to be.”

Novell sets Connector for Evolution Free

Migration motivator for *MS Exchange*-dependent businesses

Many were surprised by Novell releasing of *Ximian Connector for Exchange* (a system to allow *Evolution* to act as an *MS Exchange Client*) under the GPL. The software will be integrated into *Evolution 2.0* – due for release later in 2004 as part of Novell's Desktop Linux product.

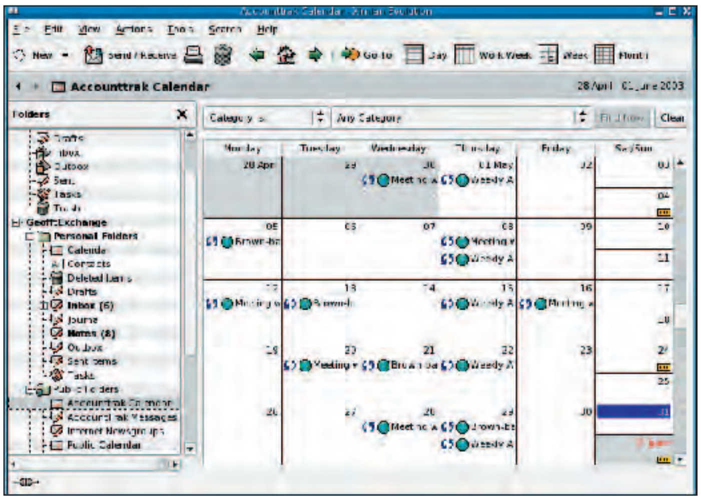
This release means the entire Evolution product is now available under the GNU General Public License. Nat Friedman, former head of Ximian and now VP of Desktop Technologies at Novell, said the release was part of a broad push to bring the benefits of Linux to enterprise users. *"Technologies like the Connector which allow Linux desktops to operate inside a mixed proprietary/ Open Source environment make choosing a Linux desktop easier for IT administrators."*

Evolution 2.0 will have advanced support for *Novell GroupWise 6.5 for Linux*, allowing users to access mail, calendars and contacts through a single client. It is also feasible that *Connector* technology will find its way into KDE's *Kontact* connectivity suite.

Novell has made a number of significant Open Source offerings designed to put its own technology at the heart of the 'Linux ecosystem'. Most significantly it now offers SUSE's native hardware/ software management app, *YaST*, under the GPL. The release of *Connector* may see one more barrier to entry for Linux on the corporate desktop collapse.

Source code for Connector is available at **http://ftp.ximian.com** and Evolution users can download Connector from **www.novell.com/products/connector/download.html**. Users will need access to a *MS Exchange 2000/2003* server with *Outlook WebAccess* enabled to use the system.

Novell's new SUSE release is available as a no-cost ftp installation from June 4th. Novell's new support strategy is based on a 'per incidence' basis and, as company spokesman Bruce Lowry said, is more in the spirit of the Open Source ethos.



Connector allows _Evolution_ users to access public calendars.

LINUX WEB WATCH

Random connections

Apropos of absolutely nothing...

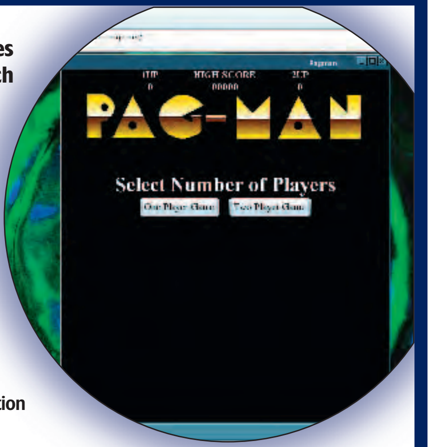
Shrek 2 **(www.shrek2.com) is one of the new breed of Dreamworks' Linux made movies. The website, which features a fairly convoluted game goes well beyond the usual 'trailers and downloads' model and works perfectly in flash enabled *Konqueror* or *Mozilla*.**

When Microsoft emerged victorious from the Browser War (having "cut off" Netscape's "air supply") it was often stated that MS's motivating fear was that software designers would bypass the OS when it came to application development and concentrate on the 'webtop'. It's not happened yet (for an idea of how the future might look see IBM's Workplace project at www.ibm.com/news/us/2004/05/111.html), but there are some pretty good attempts. *Robin* (*Remote Operating System Build in Netscape*) is one such project. Built entirely on *Mozilla* technologies such as XUL, Robin – though currently limited in scope and ability – may point to the future. The version available at http://robin.sourceforge.net features access to a small selection of XUL-based games, but also includes quite an impressive theme system. You'll need a *Mozilla*-based browser to make the most of it.

Google's weblog service (www.blogger.com) has just added proper text formatting options for users of non-MS browsers.

www.wikipedia.org, the open content encyclopedia continues its phenomenal growth and has picked up a 'Webby' for Best Community Project (www.webbyawards.com) against competition from the likes of Friendster and Fictionally. Based on the WikiWikiWeb principle, anyone can add, amend or rewrite entries without restriction.

www.shrek2.com
Don't even think about trying to see it if you're on dial-up!



robin has some surprises on the games front for such a young project!

newsbytes

■ **Blueyonder** broadband customers have received a 'free' speed boost. In response to a similar move by NTL, Telewest has upped the speed of all connections by 50 per cent. 512K buyers will see connections rise to 725K, while customers using the 2MB option will surf at 3MB/s.

■ **DireqLearn** is evangelising the benefits of Open Source software to schools in Africa with the latest release of the OpenLab thin client solution. The software is built around a Slackware core and comes with a 2.6 series kernel and KDE3.2. The group has also launched a website to accompany the software at **http://direqlearn.net/olce/.**

■ To celebrate the high-profile presence of ***Scribus*** in SUSE's new distribution, developers of the Open Source DTP package have released an update which includes nascent support for tables, extended links with external applications (especially with *The GIMP* for image editing), the ability to select text using keyboard short cuts and support for image compression in PDF output. There's also a new *Scribus* plugin available which provides facilities for exporting completed pages in a variety of bitmap formats including BMP, JPG, PNG and XPM. Brave CVS source users will have the plugin installed as standard. Download the latest release from **www.scribus.net/**

■ **Bruce Perens** has warned that costly indemnification programs for Open Source software may not be necessary for most users. Perens, a director of the newly established Open Source Risk Management consultancy, said in an interview with **IDG.com** that he would never suggest everyone required indemnification. *"I am promoting that large businesses look at their software risk. A good deal of what OSRM does is not providing indemnification, it is managing risks in other ways,"* he said.

■ A team at the National Institute of Standards and Technology has demonstrated the **fastest cryptographic system** based on the transmission of single photons. Quantum cryptography offers the prospect of 'unbreakable' codes and the NIST Quantum Key Distribution (QKD) is thought to be the first application of the technology that offers the technology useable speed (100 times faster than previous systems) and distance. **www.quantenkryptographie.at/**

Lindows to go public

Tech market recovering, but no partying like it's 1999...

Lindows boss Michael Roberts, who picked up a multimillion pay packet from the sale of **MP3.com** in the dotcom boom of the 1990s, is hoping to capitalise on the newly confident tech market to launch an initial public offering (IPO) of the Linux distributor. With Google's own IPO imminent and SCO/IBM raising the profile of Linux in the business press, industry analysts think this may be a wise move, though they caution against the unfettered optimism that contributed to the dotcom bubble. Last year the company posted a loss of US$4.1 million on revenues of US$2.1 million and the issue is expected, in part, to pay back a substantial loan made to the company by Roberts and other initial investors.

Meanwhile, routine examination of SEC filings by Lindows in relation to the IPO has revealed that the company has been involved in litigation against rival in the desktop Debian market (and former partner) Xandros. The filing states that Lindows was approached in 2001 as a potential investor in Xandros and offered to extend a loan of $750,000. The Lindows SEC filing alleges that Xandros, Inc., Linux Global Partners, Inc, Michael Bego, and William Jay Roseman *"fraudulently induced us to loan Xandros, Inc. $750,000 in exchange for convertible promissory notes, and that Xandros, Inc. then failed to repay the promissory notes when they became due."*

JPEG format comes under FUD-laden patent fire

Copyrights and patents threaten to stifle innovation yet again

The Joint Photographic Experts Group (which provides the acronym for the JPEG compressed graphics format) has reiterated that Forgent's claim of ownership (via patent) of the format is groundless. If the patent is upheld, it could mean a replay of the GIF situation, where Free software projects like *The GIMP* had to remove a feature or risk a lawsuit and fine. Two companies have so far paid Forgent more than $15 million, but the JPEG Group says there are clear instances of prior art that invalidate the claims. Also, Philips and Lucent are both lay claim to portions of the compression method in the JPEG format. Forgent's claim is based on the 1997 acquisition of a company called Compression Labs Inc, which had received a patent 10 years earlier on a "coding system for reducing redundancy." Forgent, it has been alleged, purposely sat on the patent, allowing the standard to proliferate (the basic definition of a 'submarine' patent) before making compensation demands last year. The JPEG committee suggests adoption of the superior JPEG2000 standard that, it says, is unencumbered by licensing and patent issues. **www.jpeg.org**

Imagine if people had to pay patent fees every time they wanted to look at pictures of pretty ladies on the Internet...

Jono Bacon

The founder of UK Linux, KDE developer and all-round nice guy, Jono is also a musician who's tunes have been featured on Slashdot.

COMMENT

Getting connected

"Novell has made the rather surprising decision to GPL the *Ximian Connector*. This clever nugget of software was developed and sold by Ximian to allow its *Evolution* app (which looks and operates in a similar fashion to *MS Outlook*) to connect to a *Microsoft Exchange* server. This was a major piece in the jigsaw when it was released; there was simply no complete alternative to *Exchange* on Linux, and many relied on it.

Many organisations have been ready to migrate to Linux, but have typically been held hostage by supporting *Exchange*, and although there are suitable efforts going into alternatives, *Exchange* was often a fly in the ointment. The GPLing of the *Ximian Connector* is good in two major ways. Firstly, it allows an entirely free migration to Linux at the OS and communications level. No longer must advocates extol the virtues of this free OS, applications and server package, but still try to express how a company must pay for a small and insignificant piece of software to connect to another piece of software.

It seems to me that each day, the migration path is getting cleared of chaff that stands in the way of Linux adoption. The critical mass of *"more reasons to migrate than to not migrate"* is beginning to rear its head. There is no doubt that the dedication of the many hackers, businesses, corporations and other groups is really pushing Linux as a truly viable platform.

We can expect to see more and more organisations, businesses and individuals moving over to Linux as their base and thanks must go to Ximian/*Evolution* and the *Evolution* hackers for this additional piece in the jigsaw."

CrossOver Office and CrossOverPlugin merge

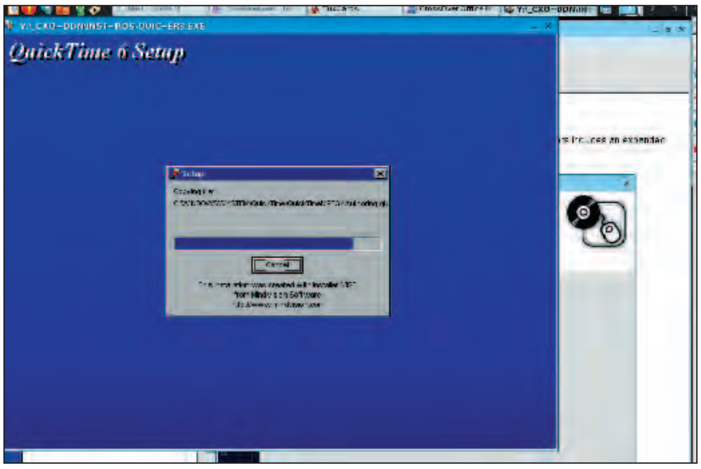
CodeWeavers notches up another triumph – more Windows apps now run under Linux than ever before!

The latest version of *CrossOver Office* from *Wine* specialists Codeweavers includes an expanded range of supported applications and also integrates the facilities for the company's other main product, *CrossOver Plugin*, now been discontinued as a stand-alone.

CrossOver Office has also been cleaved into three separate offerings. In addition to the Server application, the main application is available in Standard or Office versions. Official support now extends to more that 30 applications – including *Microsoft Office, Visio, Internet Explorer, Intuit Quicken, Macromedia Dreamweaver* and *Flash*, *Lotus Notes 6.51, Microsoft Outlook XP, and Microsoft Project.* There's also a new managed user mode enabling all users to access a single application installation.

Jeremy White, CodeWeavers CEO, said that this release cemented the products reputation as the most useable and cost-effective way of integrating Windows applications into Linux. *"CrossOver offers the cleanest, most intuitive integration with a Linux PC of any product on the market. With today's announcements, we're giving our customers more of what they've been asking for."*

He said that corporate users will get the benefits of using familiar applications in a secure, robust environment, while home users and enthusiasts get the same benefits, via the Standard edition, that remains *"in line with their budgets."*



***CrossOver Office* now comes with integrated Plugin support.**

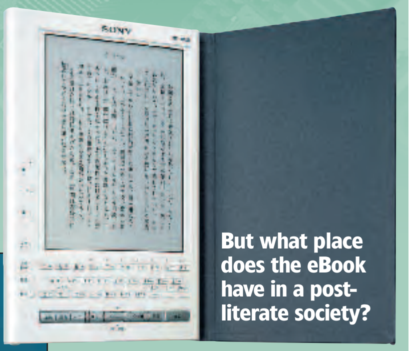
The download-only Standard Edition costs US$39.95, while the Professional Edition costs US$74.95.
www.codeweavers.com/

EMBEDDED LINUX NEWS

● i3 Micro's **Mood box** is the first Linux-based Set-top Box (STB) to be built the new Texas Instruments DSP chip which supports on-the-fly CODEC loading designed to simplify the process of regional specificity in format support. The device currently supports Windows Media Audio series 9, MP3, MPEG2, MPEG4, MPEG4-AVC (H.264), and MPEG2 HD and has facilities for IP-based TV, video-on-demand, pay-per-view, music-on-demand, Internet browsing, email, and other interactive services. The Mood box will also use an embedded version of Operasoft's latest browser for accessing Internet content.

TV and Internet convergence: maybe this time it'll work...

● While arguments rage about desktops and servers, the embedded space is hotting up with the release of the first 'paper quality' **eBook reader from Sony**. The $400 device is roughly the size of a paperback book and features a crisp screen capable of 170dpi output which, early reports suggest, is about as close to the printed page as you could imagine. The basic book is reported to have storage space for about 400 books, 50 bookmarks and a battery life good for 10,000 page impressions. There's also a MemoryStick slot for adding storage space. Sony has been in talks with leading Japanese publishers for some time to ensure that a plentiful supply of novels and reference titles available at a substantial discount compared to their traditional counterparts.

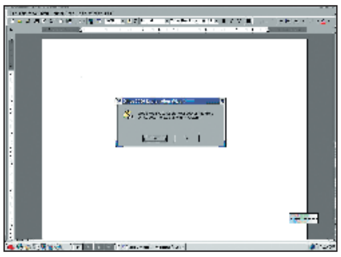


But what place does the eBook have in a post-literate society?

● Canada's on-street parking authority is planning a network of up to 800 solar-powered, wireless terminals. Each of the paystations, based on a low power StrongARM processor running a custom 2.4 kernel, will replace 12 traditional parking meters and will use GPRS to stay in contact with a central server. Unusually, the system used the *IPKG* packaging system developed for use on the iPaq Linux project.

Bridging Win and Linux?

Two projects have sprung up with the intention of bringing Windows apps seamlessly to Linux. *Project David* (in honour of 'the little guy' who smote Goliath) garnered the most publicity, though early screenshots suggest the project may owe more than a little to *Wine,* and in particular, the already mature efforts of *CrossOver Office*. A screenshot shows *David* running *MS Word*, but the scrollbars are partially obscured, something CodeWeavers' Mike McCormack says is a *COO*-only bug that doesn't appear in *Wine*.



Is *David* essentially a repackaged version of *CrossOver Office*?

HP's Linux PCs available in UK

Fine and dandy for business, but when will these tasty boxes hit the High Street?

Hewlett Packard has finally made its range of business-orientated PCs preloaded with Linux available in the UK. The new HP Compaq Business Desktop DX2000 is intended to provide *"a simple, inexpensive business computer designed to meet the basic computing requirements of price-sensitive business users,"* an HP statement said. The systems will come with either Mandrake 9.2 or Windows XP with XP Pro available as an option and in a range of configurations. All of the new PCs will also include a three-year warranty.

HP is also said to be readying notebook PCs 'certified' for use with Novell's SUSE Desktop distribution which, initially only sold in the US, will now be available worldwide.

IN MEMORIAM

On 10th May 2004, MANUEL ESTRADA SAINZ and ANDRÉS GARCÍA were killed in a tragic car accident while returning from a Free Software event in Valencia, Spain. Manuel Estrada, better known to the Open Source community simply as 'ranty', was a very active Debian developer, working on a number of projects including Lirc, VisualOS and the Orinoco_usb driver. Andrés García was a contributor to the custom Debian distribution developed for the Junta de Andalucia and a member of the Spanish Hispalinux association.

They will both be missed.

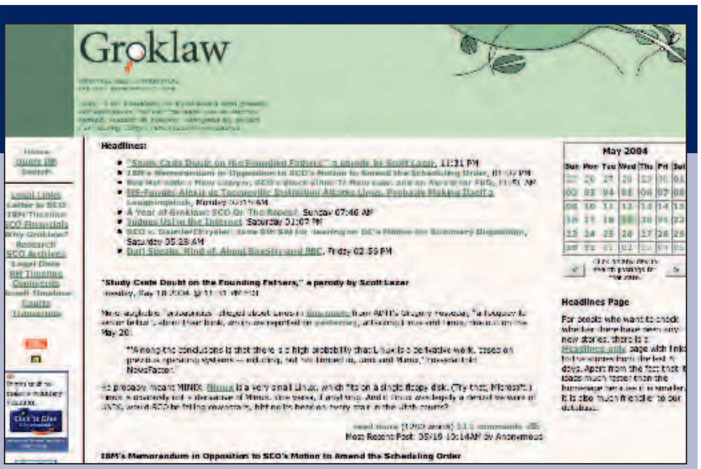
The Compaq name lives... only just!

SCO NEWS

■ Seemingly with no sense of Irony, SCO's Darl McBride has talked for the first time about Baystar's recent demand for the return of its investment. SCO's reaction to events has been along the lines that Baystar had not revealed why it was acting in this way, making it impossible for SCO to formulate a response. *Touché*. Baystar's intention, it appears, is to force a regime-change within the SCO hierarchy and to convince the management to kill the SCO Software business, turning the company into a pure IP litigation/licensing operation.

■ There has been further retraction in SCO's legal case against IBM. After recently dropping claims of copyright infringement and that the GPL was 'unconstitutional'. In fact, SCO's contractual argument against IBM now appears to be that Big Blue continued to distribute AIX after SCO terminated its licence to do so. And while that right is contested by Novell, it does raise the uncomfortable question of why the licence was terminated in the first place!

■ At the time of writing SCO's share price was heading towards the five-dollar level. At the beginning of the year, the price achieved a high watermark of over US$15 – at the start of the case it was languishing around the $3 mark. Novell, under the management of Ray Noorda (now a board member of the Canopy Group), apparently used Canopy as a front in its legal battle against Microsoft. According to Canopy, Novell sold DR DOS to Canopy under the proviso that the latter undertook a suit against Microsoft, while Novell themselves stayed out of the legal action. The Novell management has been at great pains to point out the executives that oversaw this action (if it happened) are no longer at the company.



Headlines:

- "Study Casts Doubt on the Founding Fathers," a parody by Scott Lazar, 11:31 PM
- IBM's Memorandum in Opposition to SCO's Motion to Amend the Scheduling Order, [illegible]
- [illegible]
- MS-Funded Alexis de Tocqueville Institution Attacks Linux, Probably Making Itself a Laughingstock, Monday 02:15 AM
- A Year of Groklaw: SCO On The Ropes?, Sunday 07:46 AM
- Judges Using the Internet, Saturday 01:07 PM
- SCO v. DaimlerChrysler: June 8th Set for Hearing on DC's Motion for Summary Disposition, Saturday 05:28 AM
- Darl Speaks, Kind of, About BayStar and RBC, Friday 02:59 PM

"Study Casts Doubt on the Founding Fathers," a parody by Scott Lazar

[illegible]

"Among the conclusions is that there is a high probability that Linux is a derivative work, based on previous operating systems [illegible]" [illegible] told NewsFactor.

He probably meant MINIX. Minix is a very small Linux, which fits on a single floppy disk. (Try that, Microsoft.) [illegible]

read more [illegible] comments

Most Recent Post: 05/19 10:14AM by Anonymous

IBM's Memorandum in Opposition to SCO's Motion to Amend the Scheduling Order

May 2004

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	01	02	03	04	05

Headlines Page

For people who want to check whether there have been any new stories, there is a Headlines only page with links to the stories from the last [illegible] days. Apart from the fact that it loads much faster than the homepage because it is smaller, it is also much friendlier to our database.

Follow Pamela Jones's blow-by-blow SCO details at www.groklaw.net

Hoyt Duff

The co-author of *Red Hat Linux 9 Unleashed* runs a fishing pier when he's not being vociferous about users' rights on mailing lists.

COMMENT

On being bullied...

A popular email convention is to use two dashes followed by a space to act as a delimiter for the signature. Many mail clients ignore all the following text when replying to the message – very handy for trimming the irrelevant drivel found in many sigs.

But when a published set of standards exists, to attempt to call what is clearly just a 'convention' a 'standard,' is to render standards meaningless: as is done in the current version of the *KMail* FAQ. The RFC 1036 standard (www.faqs.org/rfcs/rfc1036.html) *does not mention* the delimiter. It is mentioned in a proposed *Son of RFC1036* (www.chemie.fu-berlin.de/outerspace/netnews/son-of-1036.html#4.3.2), but has *never* been adopted into the standard. This means that an email client can not implement the commonly used delimiter and still be compliant with the standard.

The *KMail* FAQ not only appears to deliberately misrepresent this, but the application itself forces this on the users instead of providing a choice. Looking at the discussion http://lists.kde.org/?l=kde-usability&m=101595854713878&w=2 they *almost* get it correct (note the date), because it *should* be possible to turn it off: it is *not* mandatory behaviour specified by a standard. This all seems motivated – from what I can determine – by the overweening wishes of the authors (who have not responded to my query at the time of writing) to advance their opinion at the expense of the truth.

I use the delimiter and I use *KMail*. I just don't like being bullied into it and what is, in effect, being lied to. Is that what FOSS is all about? I think not!

Mailserver

Share your opinions, right wrongs and demand justice by writing to *Linux Format*. Drop us a line at: *Linux Format*, Future Publishing, 30 Monmouth Street, Bath BA1 2BW or email: lxf.letters@futurenet.co.uk

★ Letter of the month

This month's winner receives a copy of **SUSE Linux Professional 9.1**

Linux for The Disabled?

I refer to the coverdisc info in *LXF53*, page 102. At the end of the first paragraph you say, *"Blindly hitting the 'Next' button at every stage is not likely to be successful."* With all respect to the authors, that is all some of us can do: I refer to those like myself who are registered blind. Although many like me do have *some* sight, we do need access software (and hardware) to give us large print, speech or Braille. Our plight, and the small amount of such software in Linux, is not helped by magazines that pay no attention to such needs in their reviews. When was it that you last mentioned accessibility in general, or visual accessibility in particular in any of your reviews of Linux distributions, or other software for that matter? I can tell you: a very, very long time ago.

This is a most serious omission on your part, not least as the absence of access to software ensures the non-selection of such software by American government departments. One of the clauses of the *ADA* (*Americans with Disabilities Act*) makes it mandatory for all government contractors to make their products fully accessible to disabled people. It is to our shame that we in this country make no such stipulation in our *DDA* (*Disability Discrimination Act*).

Your magazine makes a big thing about freedom, *eg* freedom to use Open Source software. What about making a bigger issue of the freedom of disabled people to use such software? How about ensuring that a prime criterion in criticising software is its accessibility to people with all types of disabilities, and applying this criterion also to hardware where relevant?

In your review of Mandrake 10 in the same issue, no mention was mane of whether or not it includes accessibility software, or how good it was. You could at least have said that it included GNOME 2.4 (or later), and that this included *Gnupernicus* and *Gnuspeech*, *etc*, but no! Not a word was mentioned on the subject.

Should it not be a cause of some shame that so very little attention is given to the needs of your disabled readers? How about contacting the British Computer Association of the Blind for help? Come on! How about making *Linux Format* the one monthly computing publication that shows its real care and concern for those with disabilities? How about setting an example for others to follow, as you already do in so many ways? How about even a regular feature on computing and disability, centring on Linux, of course?!

After that, I'd better say how much I do enjoy your magazine. Thank you for a publication of real quality. However, like all things human, it can always be improved!

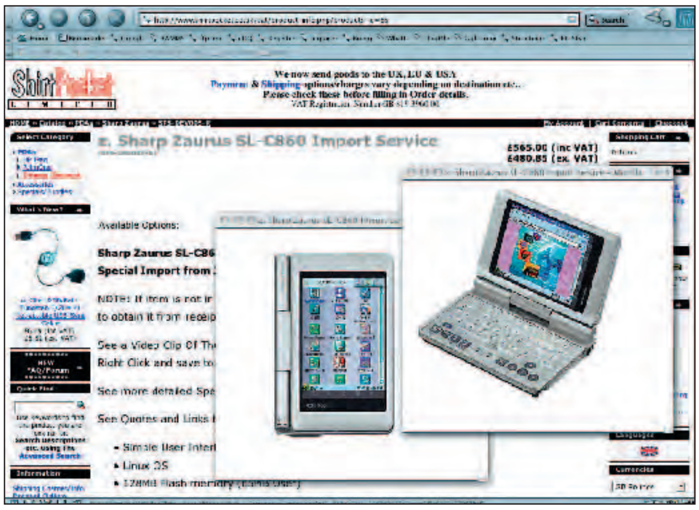
Phil Coleman, *Lancaster*

I don't think we are indifferent to the plight of those with disabilities. We have in the past run several features on accessibility issues – including a look at how SUSE can be installed using a Braille display – and as you mentioned, on GNOME accessibility tools. We actually have another article on accessibility coming up in the near future, but mostly we haven't covered the accessibility sphere recently because not a great deal has changed since our last report. I can only promise we'll try to be more inclusive where appropriate in future. In the meantime though, we'll be sending you a box of SuSE 9.1, probably the most accessible distro you'll come across.

Bloomin' Norah!

The other day my wife came home with the last CD from Norah Jones and asked me to copy it, so she could listen to it in the car, while the original stayed at home. No problems I thought, and inserted the CD in the DVD-drive on my computer. The drive went in tilt! A closer look at the CD cover showed that the CD was Copy Controlled and the Phillips/Sony Audio CD label was missing. Hmm – I remembered that I had read something about that in an issue of LXF49. Reading that issue helped me to understand that the DVD-drive would have problem with the Copy Control. I swapped the DVD-drive to an old CD-drive and used *cdparanoia* and I got the CD ripped. But I noticed that *cdparanoia* had a lot of `Unreported loss of streaming/other error in read` messages when it read the first and second track, and also some `Uncorrected error/skip` Now *cdparanoia* does a great job and did get the CD ripped and I created a CD for my wife's car.

Afterwards, I have some doubts



We now send goods to the UK, EU & USA
Payment & Shipping options/charges vary depending on destination etc..
Please check these before filling in Order details.

Sharp Zaurus SL-C860 Import Service

£565.00 (inc VAT)
£480.85 (ex. VAT)

Available Options:

Sharp Zaurus SL-C86
Special Import from

NOTE: If item is not in
to obtain it from receip

See a Video Clip Of Th
Right Click and save to

See more detailed Spe

See Quotes and Links

- Simple User Interf
- Linux OS
- 128MB Flash memory

Sharp Zaurus SL-C860 runs on Linux, and is now available in the UK!

about the CD I created. Is the CD now a better quality Audio CD than the original one? I have these doubts because of a Phillips statement, maintaining that Copy Protected CDs do not follow the Audio CD standard, and will probably have a shorter lifespan than a standard Audio CD, because of the corrupted correction codes in the first tracks on the CD.

What purpose did it serve that EMI had Copy Protected the CD? My wife bought a product at a lower quality level than a standard product. I got the CD ripped and maybe created a CD at a higher quality level than the original. Some of my Windows friends would probably have problems with ripping the first track of the CD, as they do not have a so insistent a program as *cdparanoia* to rip CDs. Some others would not be able to rip the CD at all, because they have thrown away that old CD drive.

The conclusion is: keep your old CD drive. The big producers of Audio CDs have promised even more Copy Protected CDs in the future. You should create a copy of any Copy Protected CD you have, as it properly will not have as long lifespan as the standard CD you create.
Henrik, *via email*

You appreciate that we aren't printing your surname so that you won't get into trouble with the RIAA or its ilk! And while we're on the subject, nothing printed in *LXF* either written by paid writers or representing any opinions sent in by readers constitutes an endorsement of practices that in some areas may be seen as contrary to local laws. Our official position: *if the law is wrong we should change it, not break it.*

As you mentioned, the subject was covered quite thoroughly in *LXF51*, and Andrew, another reader sent us in some other CD copying tips to *LXF53*'s Mailserver as well. We suspect that this won't be the last time Digital Rights Management issues and Linux collide!

At the Sharp end

I hate to tell you, but the information in *Embedded Linux News* on p10 of *LXF52* regarding Sharp's Linux-powered PDA, the SL-C860 is just a little bit, well, slow and inaccurate. *"Rumours... that early examples should be available soon."*?

I bought my C860 several months ago through Dynamism, one of several import companies (in the USA), and I was not – by far – one of the earliest users to do so. The C860 actually became available in late 2003.

Perhaps the writer of this bit meant to say *"available from Sharp outside of Japan without official support"* since, for the moment, the C860 has to be imported and is not officially recognized by Sharp other than in Japan.

By the way – The C860 is magnificent. There are already rumours, with some support but no exact time frame, of a C960! Drool!
Bob Weeks, *Miami FL USA*

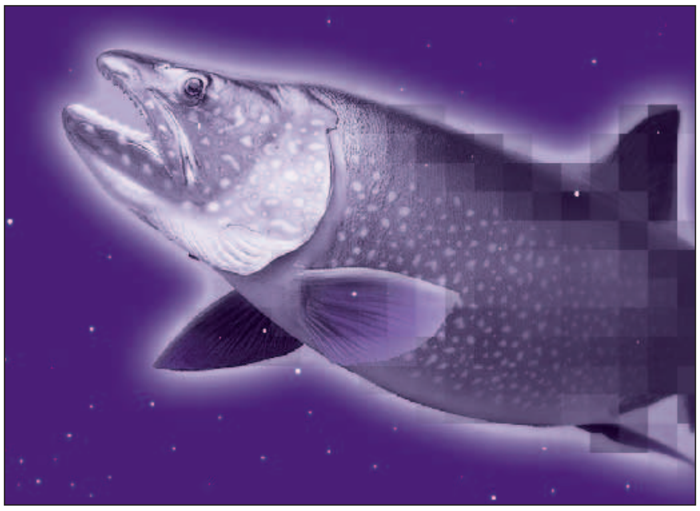
Thanks for the clarification! European readers who are interested in getting hold of this little wonder through an official supplier should visit **www.shirtpocket.co.uk/cat/product_info.php?products_id=85**. If there is a reader who has one of these who would be interested in writing about it for *LXF*, please drop us a line at **linuxformat@futurenet.co.uk** with "Zaurus C860" as the subject-line. We haven't ever reviewed 'unsupported' hardware in the magazine, but we'd be happy to include some information on such devices in a future issue, either in the form of a review or feature.

"We haven't ever reviewed 'unsupported' hardware in the magazine, but we'd be happy to include some information on the C860 in a future issue..."

Fishy fans

Just a quick note to say that I enjoyed following the SDL tutorial in *LXF52*, your April issue. I've been playing *Eternal Lands* lately, which I found out about in a previous edition of the magazine; I think it was included on a CD within your March edition. *Eternal Lands* has led me to become interested in SDL programming, so it was great to see that *LXF* also provided a tutorial for the SDL libraries.

Please keep providing this tutorial. I'm looking forward to the appearance of the trout in the source code!
Nick Faiz, *Sydney, Australia.*



Any comments or questions about our tutorials? Please send them in!

First I must congratulate you on producing an excellent magazine – I have learned a lot about Linux from them and enjoy trying out the programs and distros you include on your cover discs.

I always read with interest the tutorial section, Which brings me to the reason why I'm writing. I followed the new tutorial in LXF52 – *Trout Wars*, the game programming tutorial using SDL. I was looking forward to continuing the exercise in *LXF53*, but it appears to have vanished!

I realise you only have a limited amount of space in each magazine and that you have decided to drop the tutorial from this month's issue to accommodate something else, but I hope you are not going to leave it here. Can you please continue in

»

SUBMISSION ADVICE

WHAT WE WANT:
- Letters about the magazine or Linux in general
- Constructive criticism
- Your opinions
- Concise points about relevant subjects

WHAT WE DON'T WANT:
- Technical questions – direct those to our Q&A pages!
- Random abuse
- Nonsense rants
- 200 pages of meandering diatribe

WRITE TO US AT:
***Linux Format*, Future Publishing, 30 Monmouth Street, Bath BA1 2BW** or email: **lxf.letters@futurenet.co.uk**

READER TIPS

ANOTHER CROSSOVER FAN

After reading Eugene Meenan's tip *Cross Over to CrossOver* (*Mailserver*, *LXF53*) I just had to write in to agree. I first tried the *CrossOver Office* demo in September last year and was so impressed with the ability to run *Office 2000, Adobe Photoshop* and *Adobe Illustrator* (even though it's not officially supported) that I paid for, and downloaded, the full version.

This prompted me to try *WineX* to see if I could port more Windows software (or rather games!) across and therefore make my Windows box obsolete. With the games I installed running faster and smoother than under Windows (I must point out that I swapped a lot of my Windows hardware onto my Linux machine for benchmarking – including the GeForce4 Ti and SB Live Card) the decision to dump the Redmond bloatware was a no-brainer, and I haven't looked back since – using SUSE 8.2 for my everyday computing tasks – from simple emailing to web development to hours of happy fragging on *UT2004* (Linux native, of course).

I share Eugene's sentiment (and no doubt that of the entire *LXF* readership) that anyone thinking of making the switch to Linux could do a lot worse than trying out both *CrossOver Office* and *WineX* – I've even persuaded a friend of mine to do a lot of his MCSE on Linux using *CX Office* and *Visual Studio*!

Oh, and yes – once you've tried *Mozilla Firefox* (either Linux or Windows), you won't go back...
Tony Mottershead, *via email*

Thanks for sharing your experiences. The level of support for Windows software running under Linux is quite impressive these days. We are planning a feature on migration in an upcoming issue, and would welcome any more comments or advice that readers have on the subject – please send your pearls of wisdom to the usual address. Please turn to our *News* pages for details of some exciting new developments at CodeWeavers!

ONLY SMARTD HAS THE ANSWER!

I was just looking at your response to Gary's *Star Question* in *LXF53's Sysadmin Answers* section with regard to monitoring a software RAID set and your example script.

While the script does look like it will work (I've not tested it myself and I realise that Hans said it was a *"quick script"*) it does seem to be a little inefficient and as it stands could cause other issues.

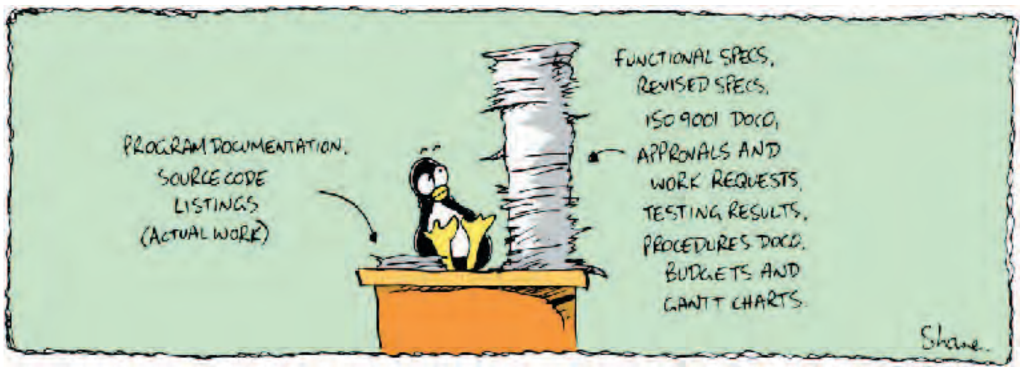
Granted, you will have an immediate notification that you have a disk failure, but the script will also be a bit of a CPU hog as it is a continuous loop; and if you're unlucky enough to have another

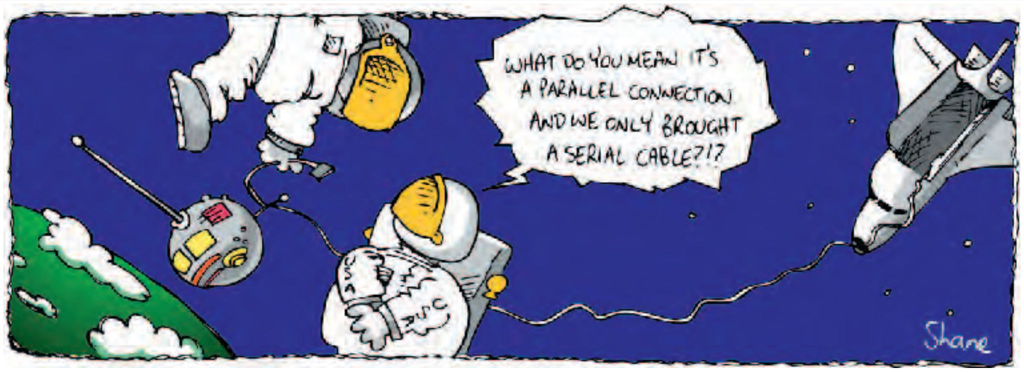
« *LXF54*? All I have at the moment is a blank window on my screen!
Graham Whiteside, *via email*

As you will see from this month's issue, we are going to continue the *Trout Wars* game (to the bitter end) tutorials, and more excitingly, when you have got to grips with the basics of SDL in this game, we have another secret project lined up. As ever, if you have any ideas for tutorials, send them in – we'll be only too happy to consider them. LXF

Helpdex

shane_collinge@yahoo.com


PROGRAM DOCUMENTATION, SOURCE CODE LISTINGS (ACTUAL WORK)
FUNCTIONAL SPECS, REVISED SPECS, ISO 9001 DOCO, APPROVALS AND WORK REQUESTS, TESTING RESULTS, PROCEDURES DOCO, BUDGETS AND GANTT CHARTS.
Shane


WHAT DO YOU MEAN IT'S A PARALLEL CONNECTION AND WE ONLY BROUGHT A SERIAL CABLE?!?
Shane

disk fail, you could also end up being mail-bombed by the script sending loads of emails to the admin account. If this is a local account, then you could have additional disk space issues or put additional strain on the network and *mailq*.

The script would be better perhaps including a *sleep*, or activated as a *cron* job once every five or so minutes (if that is the need). You could also add raid information using the *lsraid* tools for all RAID configurations.

Alternatively, you could look at using *smartd* to monitor the disks as most current disks seem to have SMART capability. The *smartd* daemon also seems to have email capability and can monitor a number of things including temperature and number of errors. I use *smartd* to monitor some Linux systems, and so far it seems to be quite stable.
Julian Parker, *via email*

Thanks for pointing out some of the drawbacks to the script presented. We're always happy to admit there's room for improvement! Thanks for the tip about *smartd* too – this looks like a subject that may be something worth writing about at some point...

CHEEKY SLAX!

Thank you for the copy of Slax on the *LXF54* DVD. (Yes, I agree with the *"cheeky"* suggestion you made in *LXF54's Mailserver* – it *is* worth getting your awesome DVD). I am always grateful for things Slackware, especially as I am on dial-up.

The most common complaint in your Live Distro *Roundup* was the need to keep the CD in the drive. With Slax you *do not have to* (yippee!) If your system has 200MB+ RAM, you specify:

```
slax copy2ram nolock
```

at boot. (Just press F1 at the boot prompt to get all those lovely options). Slax will run from RAM. Not only does this free the CD drive, it runs very fast as well. Slax is typically Slackware – you have to do a little work sometimes: I had to insmod the SCSI driver for my system after boot, and write /etc/fstab entries. Does that sound hard? I have been using Linux for less than a year – it *can* be done by a newbie! In return for your efforts you get a fast stable reliable system. Knoppix was my way into Linux, but now I am totally sold on Slax as the most usable Live distro I have ever tried.
Guy Platts, *via email*

FEDORA CORE 1

I can well understand Cliff Deamer's dilemma in trying to install the Linux operating system (re *Welcome to Linux*, *Answers LXF53*, May 2004) as I have been in the same situation a number of times. I would recommend to Cliff that if he requires a Linux system that is easy to install, he visits http://fastdiscs.com on the Internet and get a copy of Fedora Core 1 for £7 including postage.
Stan Graves, *via email*

Or indeed, he could get in touch with our backissues department, and get an issue of *LXF49* – details on page 97 – there's Fedora Core 1 on both the CD and DVD versions of the magazine!

FAILSAFE JIGDO

To avoid problems with permissions running *mkiso* or *jigdo* from the DVD. Copy *jigdo* to your home directory In a terminal, cd to *jigdo*, Run ./jigdo-lite

At the prompt, you should enter the path to the .jigdo file *ie*: /mnt/cdrom/Distros/xxx/xxx-disc1.jigdo (where xx is the specific name of the distro and file)

The resulting ISO image file will be in the *jigdo* directory. This system will of course allow you to convert a .jigdo file from the Internet. In this case, just type in the URL at the prompt.
James Baldwin, *via email*

Reviews

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

LXF VERDICT EXPLAINED

Each review is accompanied by a *Linux Format Verdict* to help you to assess the product at a glance (it's no substitute for actually reading the review, though). We award scores out of ten in the following categories:

Features: Does it provide the functions you need? Is it innovative?

Performance: How well does it do its job? Is it fast and reliable?

Ease-of-use: Is the interface well designed? Is the documentation well written, helpful?

Value for money/Documentation: Whichever is most appropriate!

For those who like numbers, the *Linux Format Rating* is a score out of 10 summing up the overall excellence of a product. It will usually, but need not be, an average of the above categories. We award scores as follows:

●●●●●●●●●●

10 The close-to-perfect product

●●●●●●●●●●

8-9 Good, but has a few niggles

●●●●●●●●●●

6-7 Does the job, but needs work

4-5 Average.

●●●●●●●●●●

●●●●●●●●●●

1-3 An utter disaster. Back to the drawing board!

THE TOP STUFF AWARD

If we really, really like something – we really think that a particular piece of software, hardware or any other sort of ware is the best stuff around – then we'll give it our *Top Stuff* Award. Only the very best will be chosen. It's not guaranteed to all products that score highly.

WHAT'S NEW...

Neverwinter Nights

Hack, slash, and magic your way through *Dungeons & Dragons* on Linux, cooing merrily at the special effects *en route* **p18**

GNOME 2.6

Spatial navigation and a new file chooser all rolled into one – can it compete with KDE 3.2? **p20**

Leylines

Strategy game seeks player with US$15 burning a hole in their pocket. It'll certainly be a unique experience, oh yes... **p22**

Gigabyte SR147

Xeons aren't dead yet – or at least that's what Gigabyte would like you to believe – judge for yourself **p24**

Maya 6

Graphics powerhouse with nothing to prove hits all-new heights for a low price, but is it worth the upgrade? **p26**

27

VIA CL10000

"It's a PC, Jim, just not as we know it" – Mini ITX evolves **p27**

Storix SBA 5.1

Backup your data and restore your confidence in just one package **p28**

Book reviews

Wil Wheaton sets phrases to stun with *Dancing Barefoot*, plus *AI Programming Wisdom 2*, the *Sendmail Cookbook*, and the new *Official Fedora Companion* from Red Hat Press **p31**

LXFBENCH 2004 EXPLAINED

To comprehensively test the capabilities of machines we review, we have developed *LXFBench 2004*: a new benchmark suite designed to push hardware of all shapes and sizes to its limit.

The test is broken down into four distinct parts: multiprocessing, uniprocessing, RAM, and hard disk, of which the first two are largely similar. The multiprocessing test creates four child processes in order to take advantage of SMP hardware, then proceeds to run *oggenc* to encode a large audio file to Ogg format, uses the *GD* image library to resample a complex image several times, and also runs an external C program to calculate the hashes of random numbers. The *uniprocessing* test is identical except that it runs on just one CPU.

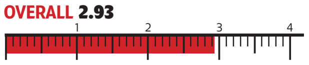
Both the *RAM* and *HD* tests use the *SQLite* database library to manipulate database information in RAM and on the hard disk respectively.

The overall score is an average of all four tests, and is presented as a bar graph for ease of reading. A score of 1 means that the machine has equalled our yardstick machine – a 1.8GHz Pentium 4 with 512MB RAM and an IDE hard disk. A score of 2, therefore, means that a machine has completed our tests twice as fast as the benchmark. The majority of the code was written using PHP 5, with the CPU-intensive tests written in C.

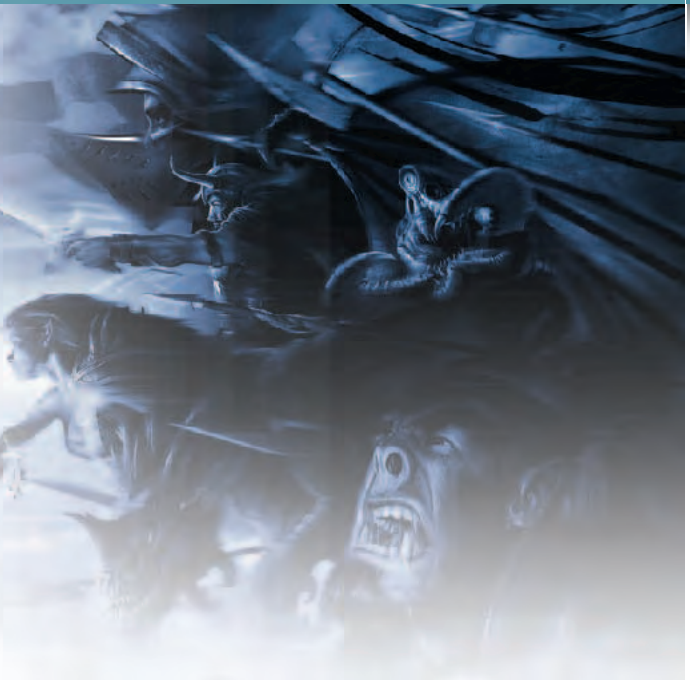
BENCHMARKS

CPU	6.07
SINGLE	3
RAM	2.17
HD	0.46

OVERALL 2.93



All our benchmarks, unless specifically noted otherwise, are run on a fresh installation of Red Hat Enterprise Linux 3 AS for the specific platform. All source code, including PHP itself, is compiled using *GCC* unless otherwise noted. The *mhash* library, created by Nikos Mavroyanopoulos and Sascha Schumann, is used for data hashing.



ROLE-PLAYING GAME

Neverwinter Nights

Here be dragons: Paul Hudson gets to hack, slash, and plunder: not that he doesn't spend too much time hacking already...

BUYER INFO

Dungeons & Dragons-based role-playing game. If you're less graphically inclined, there's always *NetHack* to consider...

- **SUPPLIER** Bioware
- **PRICE** £34.99, Add-on pack 1: £19.99, Add-on pack 2: £19.99
- **WEB** www.bioware.com/

The *Eye of the Beholder* series was incredibly popular all those years ago, and *Pool of Radiance* had its following in the past, but *Neverwinter Nights* is more than just a code implementation of several *Dungeons & Dragons* rule books, although you will find a comprehensive implementation of the rules and monsters here. Instead, the aspect of *NWN* that had everyone champing at the bit was its multiplayer option – rather than teaming up with other players to work your way through a game, one player can take the role of Dungeon Master (DM) and actively lead the adventure.

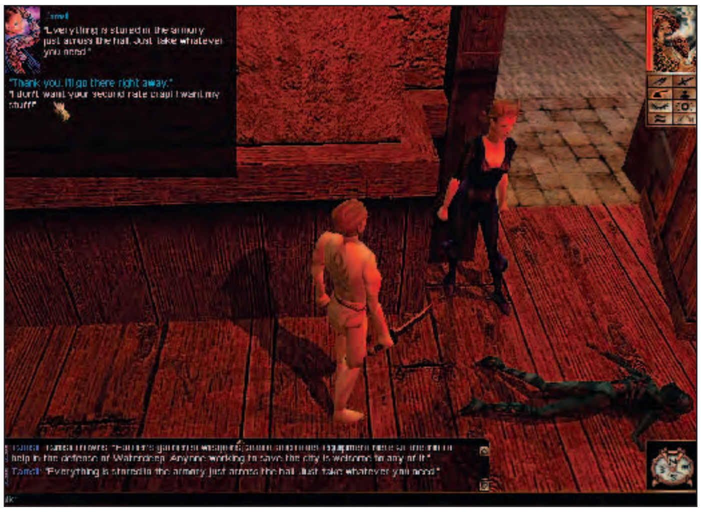
Mastering dungeons

Creating a whole adventure sounds like hard work, but *NWN* makes it straightforward. Traditionally a DM would need to sit down at a table full of books and paper, planning out maps and non-player characters (NPCs) that fill in story and help the players. Doing the same in *NWN* is theoretically easy – there's a friendly GUI interface that lets you script everything with little hassle. I say "theoretically," because sadly the DM adventure creation tool wasn't ported to Linux, which means you can play in custom games created by DMs running on Windows, but can't create your own on Linux.

The main *Neverwinter Nights* game itself comes on three CDs with a helpful manual to guide you through setup and playing. This is helpful not only to teach you the controls, but also to explain some of the more complex *D&D* rules – although you can pretty much ignore them and play the game as it comes if you like, there's a lot more depth there if you really want to get engrossed. The original adventure (it's one long story-line you follow as your character develops) will last an average player about 100 hours, although this will shorten if you have little time for what people say. There are also two add-on packs available, *The Shadows of Undrentide* and *Hordes of the Underdark*, both of which add new scenarios to the original game that extend it by 20 hours apiece, although both of them require you to start a new character to play so that you don't have an unfair advantage.

In single-player mode, the game is fairly engrossing – you do a lot of wandering, but the story that unravels as you play is consistent and strong. Having said that, sometimes it seems more like the original game was more of a proof of concept: the 100-hour adventure drags in many parts and really is just a technology showcase. The add-on packs add the good stuff – heroic characters and well thought-out plots that back up the attractive graphics as they deserve.

The graphics themselves are excellent, and its clear the developers

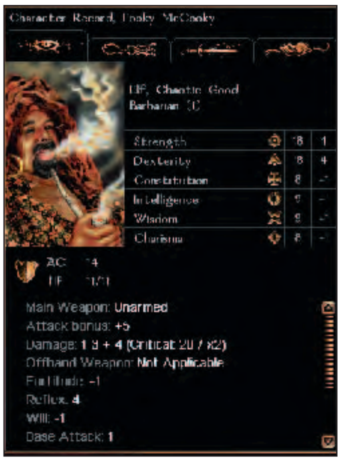


Tamel
"Everything is stored in the armory just across the hall. Just take whatever you need."

"Thank you. I'll go there right away."
"I don't want your second rate crap! I want my stuff!"

Tamel: Tamel frowns. "Father's garrison weapons and other equipment here so the men can help in the defense of Waterdeep. Anyone working to save the city is welcome to any of it."
Tamel: "Everything is stored in the armory just across the hall. Just take whatever you need."

You have free control over the camera so you can always get the best viewpoint for on-screen events, whether you're battling or conversing.



Character Record, Tooky McCooky

Elf, Chaotic Good
Barbarian (1)

Strength	18	1
Dexterity	18	4
Constitution	8	-1
Intelligence	9	-
Wisdom	9	-1
Charisma	8	-

AC 14
HP 11/11

Main Weapon: Unarmed
Attack bonus: +5
Damage: 1 3 + 4 (Critical: 20 / x2)
Offhand Weapon: Not Applicable
Fortitude: -1
Reflex: 4
Will: -1
Base Attack: 1

Pick your character carefully – you have complete control over every aspect. An over-fondness for pipe-weed might not be a good start!

MINIMUM SYSTEM REQUIREMENTS

Pentium III 800MHz; 128MB RAM
Hard Disk Install: 1.5GB; 8x CD-ROM
32MB video card with Hardware T&L
Multiplayer: LAN with TCP/IP protocol; or 56KBps Modem (2 players max)

have gone to town with the spell-casting special effects, as explosions, sparkles, fires, and more grace your actions while you battle your way through armies of monsters. The creatures themselves show a large amount of diversity – big cats, for example – come in the forms of jaguars, panthers, leopards, and various fantastical varieties that keep you guessing and learning all the time.

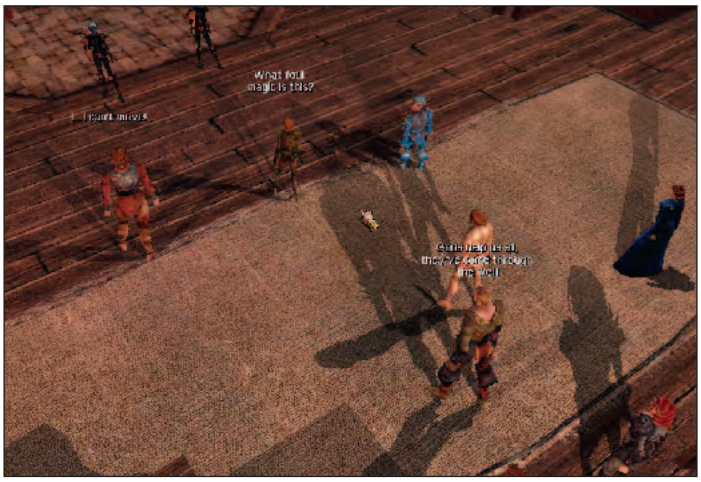
The Linux port

Unsurprisingly, we *LXF*ers like games being ported to Linux – not only does it strengthen the platform as more people use Open Source tools such as *SDL*, but it also removes another barrier to people who want to migrate wholesale from Windows. Sadly, unlike *Unreal Tournament 2004, NWN* doesn't come with its own Linux installer – the official install method is to download what it essentially the content of the CDs (all 1.2GB of it) from Bioware's website. Using this method, all you essentially buy is the serial number that validates it as original. Installing the add-on packs is a little more painful – you need to hand-delete some files, unzip others, *etc*, until you eventually have a working version.

Fortunately there's an unofficial installer available that works as you'd expect – pop the CD in, run the installer, and play the game. The reason

Wizz! Bang! Zap! The magical eye candy is top-notch, and used liberally throughout.

Eaten a particularly explosive curry? Never mind, a Paul Daniels Special should help you out!



Every player realistically casts shadows – from more than one light-source in places – so consequently you'll need a fairly muscley graphics card.

Bioware hasn't done this itself is allegedly because on an agreement with InstallShield, the manufacturer of the Windows installer, that Bioware may not allow end users to install the CD version of the game using anything other than InstallShield. The unofficial installer works smoothly, and we had the game working in under ten minutes.

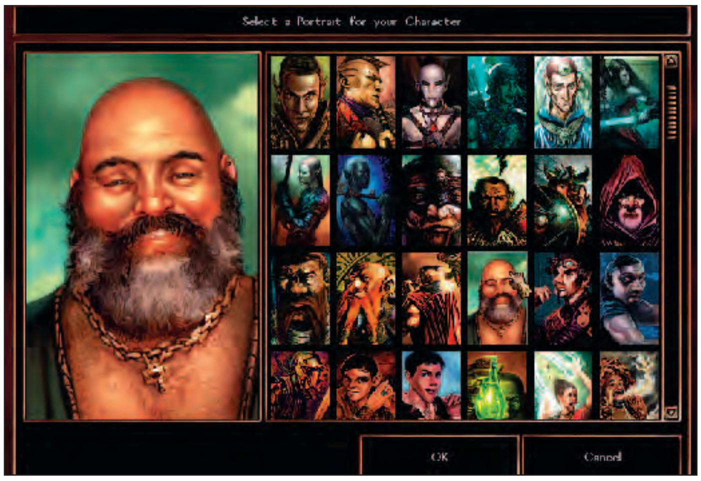
Owing to the way it was ported (Bioware promised a Linux port back in the early stages), the game itself looks identical in Linux as it does in Windows – lots of graphics, smooth multiplayer, detailed and immersive sound, and the odd confounding puzzle here and there. Happily, Bioware also produces Linux client patches along with the Windows releases, which means you get all the latest fixes and features at the same time as your Windows counterparts.

Forgotten realms

Once you're in, the action is thick and fast – you get a short introduction to the controls, explaining how camera movement and fighting works, but then you're thrust directly into the story line fighting off invaders to your castle. Controlling your player soon becomes second nature, although attacking is quite clumsy at first.

Pressing space pauses the game at any time you want, and this is pretty much a requirement if you want to be able to fight effectively – pause, stop, think, plan, unpause, attack, and repeat. As you win more, your character improves and you add more skills, get better weapons and armour, and you make friends with other characters along the way.

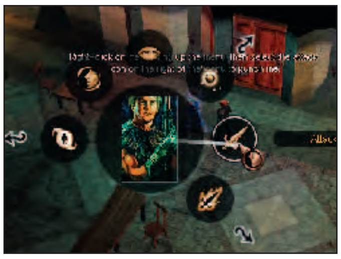
It is, however, impossible to break away on your own, as you might do in a traditional RPG – you are expected to think, talk, and act in a certain way if you want to complete the adventure, and many of the NPCs have a fixed set of lines to feed you, no matter what you've done. You can, for example, go on a massive slaughtering rampage, wiping out every innocent you can find, but then go and have a chat to some people nearby and they will be friendly and welcoming. Alternatively, try demanding five gold pieces for saving someone's lost puppy and they'll be horrified – temporarily. In this respect, it's quite linear, but it's hard to demand more: Bioware already spent five years developing *NWN* so that it could be so flexible, obviously spending more time on the engine than the story-line.



There's a huge range of character portraits available – we're wondering how much Nick got paid when he modelled for this one...?

This is quite the opposite to *NetHack*, where the graphics engine is absolutely minimal, but the saying "the developers thought of everything" is oft repeated as people try out obscure ways to solve puzzles and find they work. This perhaps explains why *NetHack* is still played today fifteen years after it was originally released – the playability has always been its strong point. That's not to say *NWN* isn't fun to play, but it does get predictable with such 'plots' as "collect four items, each of which is placed in a distant corner of the map under a large dungeon guarded by lots of creatures."

The rules are a bit overwhelming to begin with, but after about an hour you should be up-to-speed even if you've never role-played before. Things change quite drastically in Hordes of the Underdark, as once your character gets past experience level 20, all sorts of new rules come into play that allow you to perform epic deeds and vanquish even deadlier foes. This does help keep the game alive and interesting, but what helps even more is the simple fact that *Underdark* is newer and has higher system requirements – bigger and brighter special effects are littered everywhere, which gives the original game a much-needed refresh.



The Actions user interface gets overlaid onto the screen when you right-click another game object – it's clumsy at first, but necessity means that you learn quickly!

To add even more playability, there's a very active *NWN* community that produces free adventures for you to try out. The *Community Expansion Pack* adds 450 new creatures, 772 new inventory items, 1448 new scene objects, and lots more – this allows you to get newer and more advanced community-created adventures that enhance the game further.

If you want to get the best value for money, we'd recommend *NWN* plus the second add-on, *Hordes of the Underdark* – the only reasons to go for *Shadows of Undrentide* are a) that it provides some back story to *HotU* and b) if you're desperate to play more original adventures. If you keep your eyes peeled, there's a special *Neverwinter Nights* Gold pack available that comes with both the original and *Undrentide*. Let's just hope that if Bioware makes a sequel it gives us Linux folks level editing tools too! LXF

LINUX FORMAT VERDICT

FEATURES	8/10
PERFORMANCE	10/10
EASE OF USE	9/10
VALUE FOR MONEY	10/10

A good story backed up by plot-twists galore means that you don't need to be an RPG fan to enjoy this! We should really think about starting up an *LXF* adventure...

RATING 9/10

●●●●●●●●●○

LINUX DESKTOP

GNOME 2.6

Balancing delicately on a toadstool, Andy Hudson grabs his fishing rod and pointy hat to try out the GNOME way.

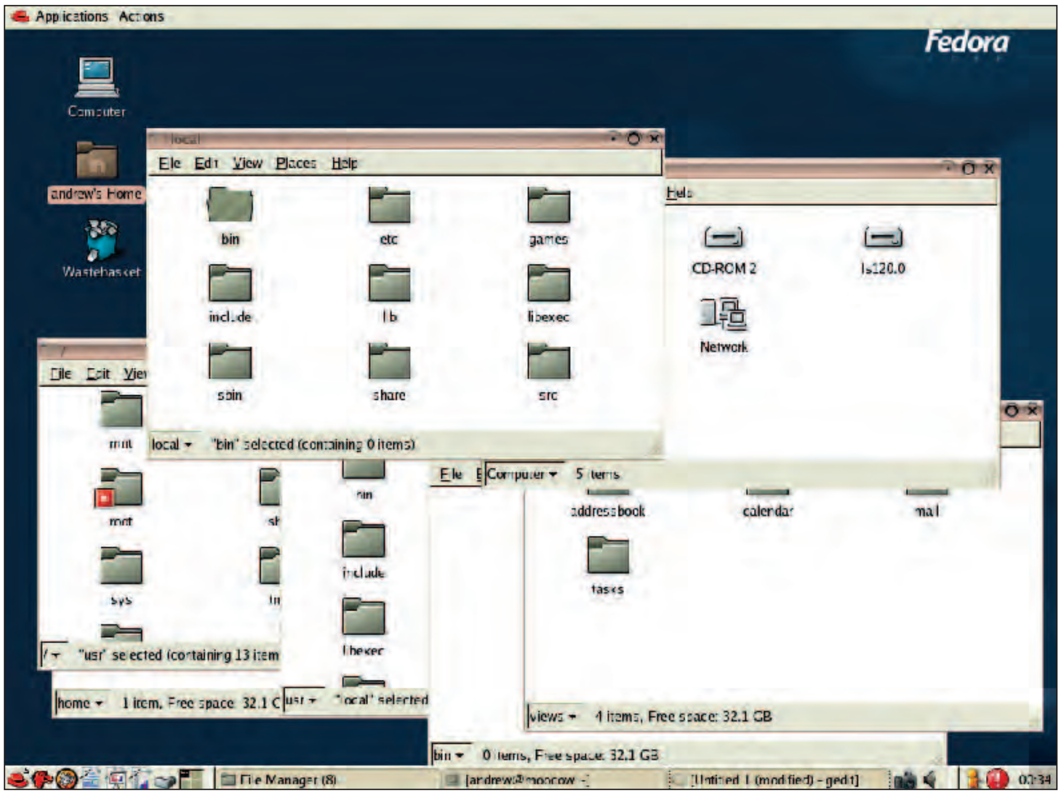
On the CD and DVD

BUYER INFO

Linux desktop environment implementation. Also try KDE 3.2, WindowMaker or IceWM.

- **DEVELOPER** The GNOME project
- **PRICE** Free
- **WEB** www.GNOME.org/

The version number 2.6 seems to be in fashion at the moment – not long after kernel 2.6 was released, we now have GNOME 2.6. With a raft of new features, and the first major release since Ximian was taken over by Novell, this is a project that now has some major backers (Novell and Sun), and as such I was pleased to be asked to take a good look at this release. A lot of effort – both paid and unpaid – has gone into making this release one of the most feature-packed yet, helped by bounties being offered for completion of a number of key objectives that has helped drive development forward. So, what's the difference and why should you consider moving to a GNOME 2.6 desktop?

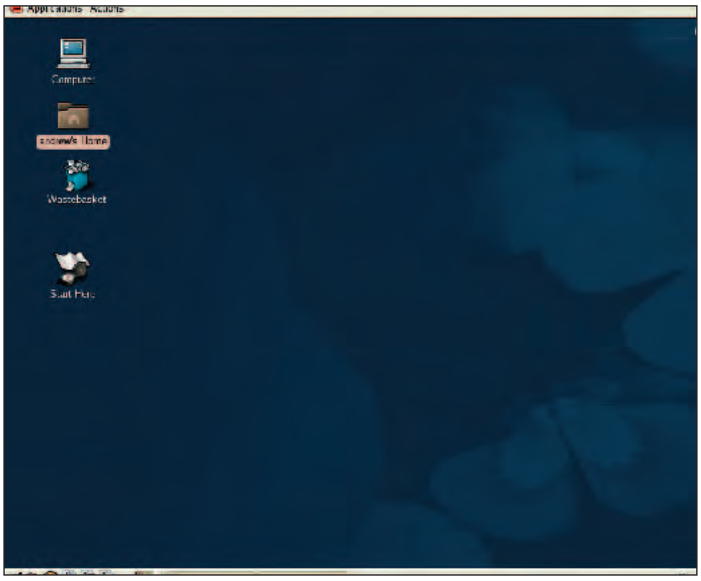

Applications Actions
Fedora
Computer
andrew's Home
Wastebasket
File Edit View Places Help
bin
etc
games
include
lib
libexec
sbin
share
src
local
"bin" selected (containing 0 items)
CD-ROM 2
ls120.0
Network
Computer 5 items
addressbook
calendar
mail
tasks
views 4 items, Free space: 32.1 GB
"usr" selected (containing 13 items)
home 1 item, Free space: 32.1 GB
bin 0 items, Free space: 32.1 GB
File Manager (8)
[andrew@moocow ~]
[Untitled 1 (modified) - gedit]
03:34

All these windows springing up everywhere, and nowhere to keep them... The taskbar sensibly groups windows together, keeping them more organised, but the ability to turn this feature off shouldn't be so obscure.

Well as soon as you log in, you can't help but notice something vaguely familiar about the desktop. It takes a couple of seconds to sink in before noticing the addition of a Computer icon. This is the first time that this kind of thing has been implemented in a GNOME desktop and it does seem a bit strange to someone who's used to using a mixture of the command-line and Nautilus to browse through the filesystem. Opening up the Computer gives you further icons for the local filesystem, as well as any optical, floppy or other storage options. It even shows a Network icon for browsing local networks. Furthermore, you can use GNOME's Virtual Filesystem support to add folders for websites as well as more run-of-the-mill Microsoft networks.

Need to upload a batch of new pages for your website? Simple, just drag-and-drop it onto the necessary icon for it to happen. For a moment, you could almost believe that you are hallucinating and actually using a Microsoft system. Is this a bad thing? I don't think so, as it will make new users feel comfortable with the system – which is one of the major hindrances of implementing Linux in a large user environment.

Nautilus has also undergone a bit of a re-think. No longer are you limited to browsing your files using one window; now a new window pops up with every folder or directory you open. Calling this a 'spatial' interface, this can nonetheless be somewhat disconcerting if you like order on your desktop, especially if you're planning to do some major browsing of the

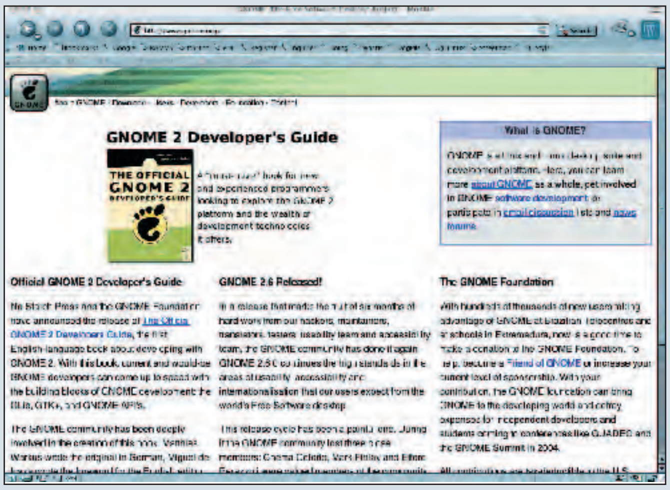


Where have I seen this before? The addition of the Computer icon certainly invites comparisons to Microsoft's way of doing things.

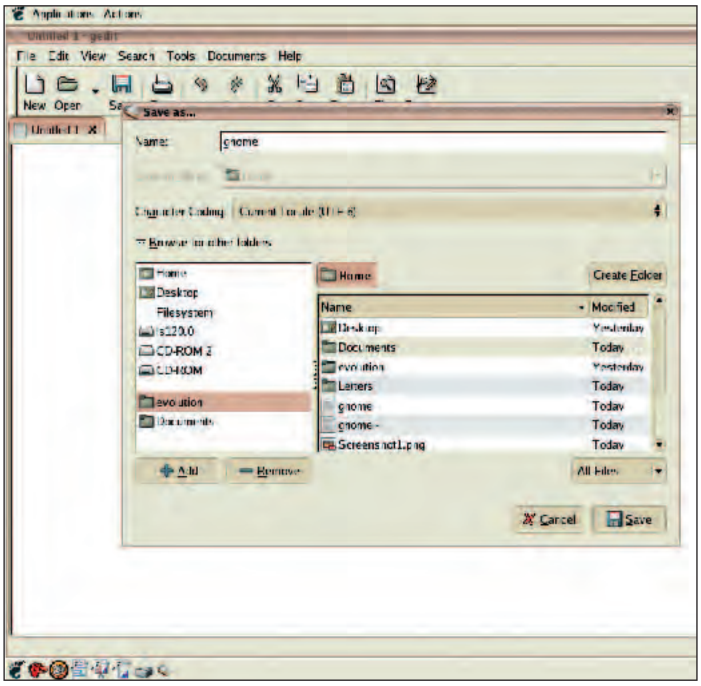
A BRIEF HISTORY OF GNOME

Gaining ground day-by-day...

GNOME came first... at least, old-GNOME did. Once KDE emerged as the first serious attempt to provide a unified desktop, various Free software pioneers (including RMS), worried by the licence restrictions inherent with *Qt* at that time, set out to produce a totally Free desktop environment. After 18 months of frenzied development and coding, GNOME 1.0 was released in March 1999 and since has become one of two major players in the desktop arena. Helped by the addition of some 500,000 desktops via Sun, GNOME is fast becoming a major part of Linux and one which more and more people are choosing to use.


GNOME 2 Developer's Guide
THE OFFICIAL GNOME 2 DEVELOPER'S GUIDE
What is GNOME?
Official GNOME 2 Developer's Guide
GNOME 2.6 Released!
The GNOME Foundation

The website is an example in clarity and design to all FOSS projects.



The file chooser dialog will feel comfortingly reminiscent for new users.

filesystem. Your screen can soon become excessively cluttered – not to mention your taskbar, which can be almost overwhelmed by all the windows that are open. The bad news is that it is not easy to turn off; you will have to delve right under the hood to tweak it in the Configuration Editor. I'm not sure about the actual benefits of having a lot of windows open at any one time – it really does remind me of working with Windows 95 (shudder). You can get round this by doing one of two things: either holding down shift while you double-click, or by double-clicking the middle mouse button. Either way, this has the effect of closing one window when another one opens, neatly avoiding excessive cluttering. On the flip-side though, the browsing does allow you to jump up directory levels with ease by using the 'parent folders' button in the bottom left of every window.

Location, location...

A nice touch to the spatial interface is that it remembers the exact location and condition the window is in when you close it. Returning to the directory in the future, GNOME will re-display it in the last known configuration. Again, if you are hunting through directories this can either be a blessing or a curse. I have to admit that it made me think of a seriously untidy office (*LXF Towers*, anyone?) as you return to the folders as you last left them. One place where it does fall down is if you try to use the List View. Instead of remembering the user-defined column widths, it instead reverts to the default each and every time you use it; which if you are used to the spatial interface, can quickly become annoying. The spatial interface has been a highly controversial addition to the GNOME desktop, so it remains to be seen if it will stand the test of time.

The CD burning qualities of Nautilus continue to be improved upon, with ISO burning simply a matter of right-clicking on the necessary file and selecting 'Write to CD'. There is also an easy way to add music files to your library, again by right-clicking the file(s) in question and selecting Add Music to Library – made even more convenient by the improvement of support for existing formats, as well as the inclusion of additional formats. For those who are familiar with Mac OS X, it'll strike them as somewhat *iTunes*-esque.

Tied in with the file manager aspects of GNOME is the new file chooser dialog box that appears when you open or save a file within a GNOME application. Again, this feels strangely reminiscent of Windows, especially when you see the listings in the left-hand pane, which are replicated from the Computer folder on the desktop. There is also the ability to add shortcuts to the list, an example being a communal documents folder on a network or something similar. This is more of a convenience thing rather than a necessity, but again it clearly demonstrates that thought has gone into GNOME to make it more accessible for the average Windows user who is switching (or being switched, depending on how you look at it) to a Linux desktop. Other things make this painfully obvious, such as the network activity panel icon or the keyboard language panel applet.

Other touch-ups

The standard stuff is all here, including a souped-up Epiphany browser. It still retains its clean lines and uncluttered approach, which has made it popular with the GNOME camp. However, with the accelerated pace of the *Mozilla* project, one wonders if *Epiphany* can keep up with *Firefox*. Various niggling problems have been tidied up, with improved auto-complete of URLs finally reaching a good standard, as well as the ability to choose fonts by point size as opposed to the more obscure pixel size.

Overall, the whole package is greatly enhanced by the use of *GTK+ 2.4*. Not only has it enhanced the overall look-and-feel of the GNOME environment, but it has also had a noticeable impact on the actual speed and performance.

As mentioned earlier, GNOME has suddenly won a lot of support from the corporate world. Not only has Novell now got a concern in what happens to GNOME, but Sun is going to be responsible for the rollout of GNOME to 500,000 desktops in China. Such mass-adoption can only bring benefits to the GNOME community at large, affording it greater coverage and development to further refine an already great desktop. It still remains to be seen what benefits can be derived from the Novell/Ximian acquisition, but whichever way you look at it, the future is seeming exceptionally bright for the GNOME project. LXF

LINUX FORMAT VERDICT

FEATURES	8/10
PERFORMANCE	9/10
EASE OF USE	8/10
DOCUMENTATION	8/10

An extremely polished effort from the GNOME camp, but there's room for further improvement – which depends greatly on intelligible user feedback.

RATING 8/10

●●●●●●●●○○

STRATEGY GAME

Leylines

On the CD and DVD

Battling elves and evil wizards, Nick Veitch examines the lay of the land...

BUYER INFO

Turn-based resource-management/strategy game. *FreeCiv* is probably the closest – and is free.

- **DEVELOPER** Game thoughts
- **PRICE** $15
- **WEB** www.gamethoughts.com/leylines/

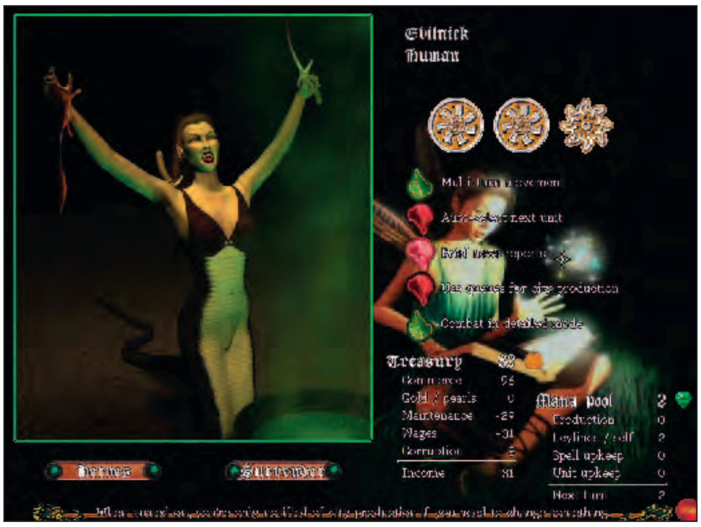
Once upon a time, there was a Unix game called *Empire*. Building units and using them to explore the ASCII-based landscape, epic battles would be fought between upper-case alphanumerics for supremacy of the virtual world. It inspired *Civilization* – in fact, the original *Civ* WAS *Empire*, but with flashier graphics and cut-scenes. Since then, the gaming world has not looked back, with hundreds of variations on this theme.

At this point I would like to say that *Leylines* is one of them. It does take the standard model – start out with a city, build improvements. Create units to found new cities, build armies, conquer opponents *et al*. For a splash of variety, there are magic spells to be used, either to protect units, destroy enemies or perform some other useful services. On the face of it then, it seems that Leylines is *Civ* with a twist?

Bread & Circuses

Well, no. Unfortunately somewhere between the concept and the delivery of the game, a few elements have been lost. There is a minimal tech tree, which decides the order in which buildings can be built and the units that can be produced. Each town has a preset ability to generate wealth, food and magical mana. Spare citizens can be allocated to any of these activities, with the appropriate results.

Of course, buildings cost money in maintenance, as do units. So, in order to fully develop your cities – while simultaneously maintaining a decent enough defence force – you'll need a lot of cash. In the old *Civ* model, this usually meant ruthless expansion - build more and more cities, run them efficiently and use the leftovers to subsidise special buildings and a standing army. This doesn't seem to work so well in *Leylines* because of the attitude of the people. Over time, they become discontent: the only way to make them more content is to have a festival, costing a sizeable wedge of cash. The more cities you have, the more often you'll find yourself doling out the sausages and beer, until there comes a time when you can't afford it. The mood in a city turns ugly, then with no other warning, the city AND all the troops it contains rebels, and is immediately lost to your empire. The economics just don't work out.



Keep an eye on character information to see how few resources you've got left after buying everyone a pint and a sausage... again.

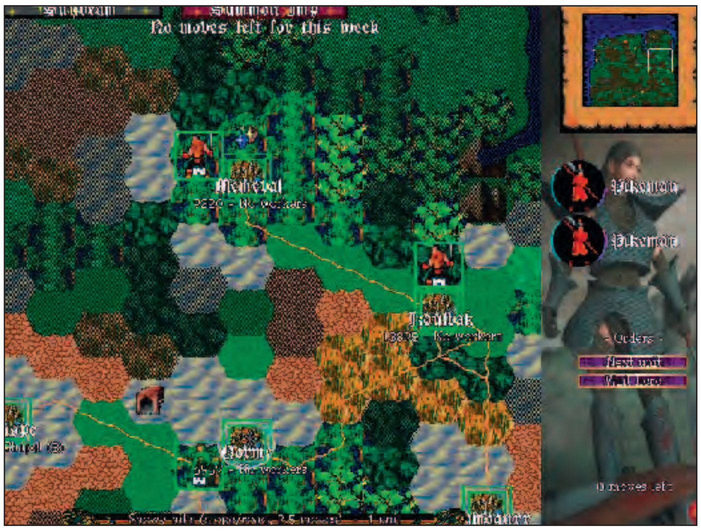
Combat is a fairly straightforward affair, with a little more detail than *Civ* in that the units taking part can have individual characteristics, and there is some concept of a formation. Even so, this isn't a particularly exciting aspect of the game.

Magic, I suppose should be touched on, but it is often irrelevant. Spells can be cast to improve the ability of units, or to attack other units, and even create units. Spells must be researched before they can be cast, and the casting itself can take some time. This revolves around the creation of mana, another resource managed by the cities. For the reasons already mentioned, this is likely to be in short supply. In the time it takes to cast an offensive spell, and enemy unit could probably have done all the pillaging it wanted to – it can take many turns to even cast the more powerful spells.

State of code

If a marginally entertaining game had slick production values, it may be due more sympathy. However, *Leylines* does not possess any such redeeming feature. The *SDL* libraries are excellent and have led to many fantastic games originating on or being ported to Linux – but sadly, *Leylines* is not one of these. Without having seen the code, we may be making presumptions, but the fact is, this game doesn't seem to have been coded particularly well. It eats up CPU time while not actually doing anything other than flash an object on the otherwise static map. Oh, and you will have to turn down the graphical detail to the minimum for the 'flashing' effects to strobe at the rate of a geriatric tortoise on a rocker switch. More amazingly, some of the graphics are corrupt. The on-screen cursor often can't keep up with the mouse movement. And we won't even talk about the sound...

Leylines is a two-time loser, then. There is the germ of a game there, but only the fact that I had to review it kept me playing this for more than an hour. Perhaps, if the code and playability issues were looked at, *Leylines II* could be great, but unless you're a really devoted strategy game completist, this really isn't even worth the time it takes to install. LXF



Hex-based play-areas tend to only be used in wargaming these days.

There's a quite limited tech tree without masses of unit types.

LINUX FORMAT VERDICT

FEATURES	4/10
PLAYABILITY	4/10
GRAPHICS	3/10
VALUE FOR MONEY	1/10

A poor game, poorly executed - definitely one to avoid. For a better *Civilization II*-esque experience, try *FreeCiv* – get your downloads from **www.freeciv.org/**

RATING 2/10

●●○○○○○○○○

BUDGET 1U SERVER

Gigabyte SR-147L

Inexpensive and re-usable servers abound in the industry today, but Paul Hudson has found one that particularly stands out...

BUYER INFO

The Gigabyte range also includes the SR-113 and SR-127. Or you could configure them by hand...

- **SUPPLIER** Upgrade Options
- **PRICE** £829.95 + VAT
- **WEB** www.upgrade.co.uk/

When the difference between a Pentium 4 and a 1-way Xeon is pretty much the place the decimal point comes in the price, it's no surprise that people who want nothing more than a workhorse box choose the former. The problem then usually comes down to how many economies do you want to make – IDE or SCSI? 1GB RAM or 512MB? The more you cut back, the more you save, but it's exceptionally hard to hit the sweet spot between price and performance to give yourself the best server for the money.

Gigabyte has provided us with several excellent machines in the past, none of which have failed to impress. Although it's easy to churn out what are essentially little more than white box units with a brand stamped on the front, Gigabyte (through its European partner, Upgrade Options) offers a selection of hand-picked favourites that come preconfigured for maximum performance whilst still weighing in with a low price. This machine came with a 2.8GHz Pentium 4 chip (533MHz front-side bus), 1GB of RAM, and twin 80GB SATA drives. SATA drives are becoming the standard now for non-SCSI drives, which is somewhat problematic – comprehensive SATA support is still lacking from many Linux distros, which meant Mandrake 10 Community edition failed to install, and even Knoppix failed to detect the drives. Fortunately, we run our benchmarks on Red Hat Enterprise Linux, which worked fine, but you need to be willing to pay the extra to get it.

Fast and loud

For such a small server, this thing really makes a lot of noise. As it's a 1U server, it's not possible to get large-bladed fans in there to keep the volume down, so instead there are six tiny fans that don't really do much except make a lot of noise. Of course, all six combined manage to make a difference, so the heat is kept down in exchange for noise. This is always a problem in 1U servers, and there's very little that can be done about it with the current architecture. AMD has recently released new, low-power Opterons that should reduce the need for fans – we're looking forward to see whether Intel will respond with something similar.

The noise is justified by the performance – keep in mind our yardstick machine is a 1.8GHz Pentium 4, so this machine is only 55 per cent faster on raw CPU power. The multiprocessing CPU score, however, is 2.0, making it twice the speed of our yardstick. This is backed up with a single CPU score of 1.82, which again places it well ahead of what we have expected. Both chips have an L2 cache of 512KB as they are both Northwood-core CPUs, so the speed difference shown here is down to nothing but excellent build quality.

Also of note is the very high RAM result – 1.95 is an excellent score, highlighting how much a difference the 533MHz front-side bus can make. This is unfortunately dragged back a little by underperforming hard disks: 0.52 isn't the best we've seen, but neither is it the worst. SATA drives as these are, this is about an average result, although if you're desperate to boost that up you can chain the two together as a striped array and you'll get extra speed.

Where to put it

Overall performance points to best deployment as a front-end server that can cache its data in RAM. With 1GB here this thing will take web serving in its stride, will plough a swathe through databases at lightning speed, and leave other file and print servers wishing they had never been born. What's more, it does so at what is simply a bargain price, thanks to a pricing strategy that Gigabyte continues to follow well.

On the flip-side, the Pentium 4 chip seems to have ground to a halt in terms of scaling upwards. The 3.06GHz chip was released over a year ago, and as we write this the first 3.4GHz Prescott chips are arriving on our desks for review purposes, which means the chip has barely moved over the last year to 18 months. As it is forward movement that drives prices down for lower-clocked chips, the lack of newer models has kept the actual CPU prices fairly stagnant, and we wouldn't be surprised if we saw Athlon 64-based devices from Gigabyte before the year is out.

Having SATA is as much as an advantage as a disadvantage, with the biggest plus being that the hard drive scores are above the IDE average without adding any cost. Furthermore, the SATA cables take up less space in the box, which helps airflow that little bit more. On the downside, driver issues abound: Red Hat Enterprise Linux works fine, but expect trouble with Debian, Mandrake, and various others. Still, it does show that Gigabyte is showing leadership in the field, as SATA is clearly the way forward – we strongly recommend you test a unit out before making a commitment. LXF

BENCHMARKS

CPU	2
SINGLE	1.82
RAM	1.95
HD	0.52

OVERALL 1.49

LINUX FORMAT VERDICT

DOCUMENTATION	7/10
PERFORMANCE	8/10
EASE OF USE	8/10
VALUE FOR MONEY	10/10

A capable and well-priced machine that brings SATA to the masses. Watch out for driver issues, though!

RATING 8/10

●●●●●●●●○○

PS/2 ports are always welcome – many users of budget servers will want to re-use old input devices.

MINI-ITX MOTHERBOARD

VIA EPIA-TC 10000

Nick Veitch scales down to review another mini-ITX motherboard.

The PSU is fine for the default configuration, but what if we want to throw in some more power-hungry components?

BUYER INFO

A great new twist on the mini-ITX design, but you may want to wait for the MII if you need a multimedia-capable board.

- **MANUFACTURER:** VIA
- **PRICE:** £139
- **WEBSITE:** www.viapsd.com/

Although other manufacturers have mini-ITX form-factor boards, it is VIA that has made this market its own – particularly with Linux users – thanks to active support of the hardware in terms of drivers. The range of options available in mini-ITX is already pretty good, and VIA has extended it further with the release of the EPIA-TC board. In fact, since the release of the TC, two more options have been announced: the PD and the MII boards – the latter we will be looking at next issue.

Cards & power

The TC board is available with either Eden (fanless) or C3 (Nehemia core) processors. With the CLE266 North bridge and VT8235 South Bridge, this means the TC delivers the MPEG2 decoding abilities of the Unichrome, but lacks surround sound. Also uncommon in the series, there is no TV-out connector, though the LVDS header allows a number of modules to be plugged in for different displays (such as direct LCD support).

The unique selling point of the TC, for the moment, is the optional Cardbus and Compact Flash card readers. Although IDE-to-Compact Flash adaptors are available, they are a little clumsy. This option on the TC board requires no extra power connection, doesn't take up an IDE slot and conveniently directs the card expansion out the backplane.

The IDE interface also supports Direct DOM (Disk On Module) devices, by supplying power. With the capability of booting from either these or Flash RAM cards, all the possibilities for an almost instantaneous power-on are there, waiting to be used.

Another useful feature that will make it attractive to embedded designers or even home-modders is that it includes an integrated 60W PSU. Instead of a jumble of power connectors, there is simply an input for an external 12v transformer.

This is a great advantage – fanless power supply units of this type can cost as much as 50 per cent of the cost of the boards themselves – so if you are trying to make a compact and quiet system, this could save you a lot of cash. For IDE drives a small header on the board provides power, with a supplied cable.

While this is a neat solution, it is also a little restrictive – it's unlikely that you will be able to make use of all the expansion capabilities of the board at once. According to VIA's own figures, the board itself consumes 14-25 Watts on its own, depending on usage.

This leaves 55W remaining for IDE devices, PCI cards, and sundries such as USB. For IDE and PCI, you can discount another five or six, as these are on the 3.3v supply. It is certainly possible to run a DVD drive and a hard drive from the onboard power, but if you want to combine this with a hungry PCI card or perhaps upgrade to a CD/DVD burner, you may face brownouts. Of course, it is possible to power the IDE devices from a separate PSU, but that rather defeats the point.

Drivers

Of all the components included in this VIA board, only the sound card and the graphics require a special mention as regards drivers. On the audio side, current kernels do have support for the VIA through *OSS*. Unfortunately, while convenient, these drivers aren't particularly good, especially on these boards where the sound tends to be very choppy. VIA provides its own audio drivers, which are certainly a lot better, but if you wish to stay in the Open Source domain – or use custom kernels – the *ALSA* drivers are pretty good.

On the graphics side, things get a little more problematic.The VIA drivers include support for the MPEG decoder which – for various reasons – is proprietary. Therefore, only closed-source binary drivers are available. These have been targeted at a number of popular distributions, but they do lag behind a fair bit (for example, at the time of writing, a driver for Fedora Core 1 has just been released, but Core 2 is due out any day). The *Lite* source doesn't include the code for the MPEG decoder. At the moment, the only software taking full advantage of the decoder is the collaborative VIA/*Xine* project *VeXP*.

Options

On the back panel, you'll find the usual keyboard, VGA and audio connectors, two USB 2.0 ports (there are additional onboard headers for two more) and an RJ45 LAN socket for the onboard Tahoe networking device. The optional Compact Flash and Cardbus module occupies the rest of the space on the back panel when installed.

VIA has done a very good job of layering this market with options – it is no mean feat to fit so many features onto such a tiny board. More integration will lead to more features available on the same board (as with the upcoming MII), but for the moment, you have to choose between features like an onboard PSU, video out, Cardbus support and so on.

This card works as well as you might expect. The processor has more than enough power for everyday tasks (and can probably handle MPEG2 without the onboard chips). The extra capabilities of Cardbus and Compact Flash may make it particularly appropriate for some developers, and the integrated PSU is a good move (though perhaps we could suggest using a 90W one next time?) LXF

LINUX FORMAT VERDICT

FEATURES	8/10
PERFORMANCE	9/10
EASE OF USE	8/10
VALUE FOR MONEY	8/10

Not the best overall, but still respectable, this may well suit particular projects where size of board is a primary concern. Compare with our MII review next issue.

RATING **8/10**

●●●●●●●●○○

BACKUP SOLUTION

System Backup Administrator 5.1

Keeping your data safe has never been easier, as Paul Hudson found out...

BUYER INFO

Backup and restore administrator. Also consider *NetVault*, reviewed in *Linux Format* issue 43.

- **SUPPLIER:** Storix
- **PRICE:** From $79
- **WEBSITE:** www.storix.com/

Storix has a long history of providing quality backup software to help home users and network admins alike keep their data safe, and this is helped along by it also providing a free, personal edition of *System Backup Administrator* for non-commercial home use. The trade-off for using the *Personal* edition is that you're stuck working with just the local host, but this is usually more than enough for home users.

Enterprise users need to turn to *Desktop Edition* ($79), *Workstation Edition* ($425) or the *Network Administrator Edition* ($795+) for more advanced features, which means there's a product to fit your budget and functionality requirements. The primary difference between the *Desktop* and *Workstation* editions is the target environment; *Workstation* is a bit more advanced – aimed at larger businesses – whereas *Desktop* falls between *Workstation* and *Personal*, and is aimed at home power users and SMEs.

The unique selling point of Storix applications has always been its recovery procedure – at its simplest, this is as easy as putting a CD in and following a wizard. As the same CD has your backed-up data and also the Storix software to restore it, you can do everything from one place and using one environment. This alone tends to win over any administrator who has had to live through disaster recovery in the past, but *Storix SBA 5.1* is far from a one-trick wonder.

New and improved

All the standard backup features are here, as you'd imagine, so incremental backups, partial restores, tape loader support, and other run-of-the-mill functionality is all here. Minor tweaks have been made to the user interface, but it's still pretty hard to figure out and not very attractive. It's important to keep in mind that *SBA 5.1* does work on Linux and AIX, so cross-platform compatibility is a must for them. Having said that, a GTK/*Qt* GUI option would make the whole package friendlier and easier to learn. This release introduces a new themes system that allows you to change the colour of the interface – this at least gives you the choice of ugly red, ugly blue, or ugly green.

The five big new features in this release are full support for back up and restore for SANs, the ability to migrate backups across media, automatic backup verification, automatic client software updates, and *LVM* snapshot. Each of these is a powerful addition, and very welcome. The backup migration and *LVM* snapshots are perhaps the best of the bunch – the former allows you to backup to a local disk (or other media) then move it elsewhere, for example to an off-site tape system. Although you could backup directly to the offsite store, it does mean the backup will take a much longer time – a local backup would take just a few hours and would minimise system downtime. LVM snapshots allow you to take a hot-copy of your LVM volume while it's in use, which again helps to cut system downtime.

The client updates were one thing we were unable to test, although it's a common-sense feature. This allows you to check for and download updated versions of *Storix SBA 5.1* through the GUI. As there were no updates available at the time of writing - presumably no bugs have been found so soon after release – this was just not testable, but it presumably does what it says.

Still the same

A big plus to this release is that it is fully backwards-compatible with *SBA 4.2* (the previous release). This even works if you boot from an *SBA 5.1* recovery CD to restore 4.2 data, which means administrators don't have to worry about compatibility checks when they upgrade – it just works, as it should.

Installation remains easy, and simply involves answering questions on the command-line. Annoyingly there is no obvious way to uninstall the program – perhaps Storix thinks no one would want to uninstall? The best we could find was to re-run the installation wizard, which automatically detects and removes previous installations, then hit **Ctrl-C** before it got around to installing the files afresh – hardly a solid solution.

Moving on – and this is perhaps what many are waiting to read – the free technical support remains intact and as good as ever. Even if you are opting for the free *Personal* edition, you get free web-based tech support, which shows just how committed Storix is to helping people get to grips with its software.

Although there are much more powerful backup solutions available, few are as widespread as *Storix SBA 5.1*. This latest release shows the platform continues to mature and also that Storix continues to have its eye on the ball. If you want to get your system safe now and don't mind a GUI that makes even *Maya*'s seem nice (see page 20 for more information), give *Storix SBA 5.1* a try – there's even a 30-day evaluation version available for free download. LXF

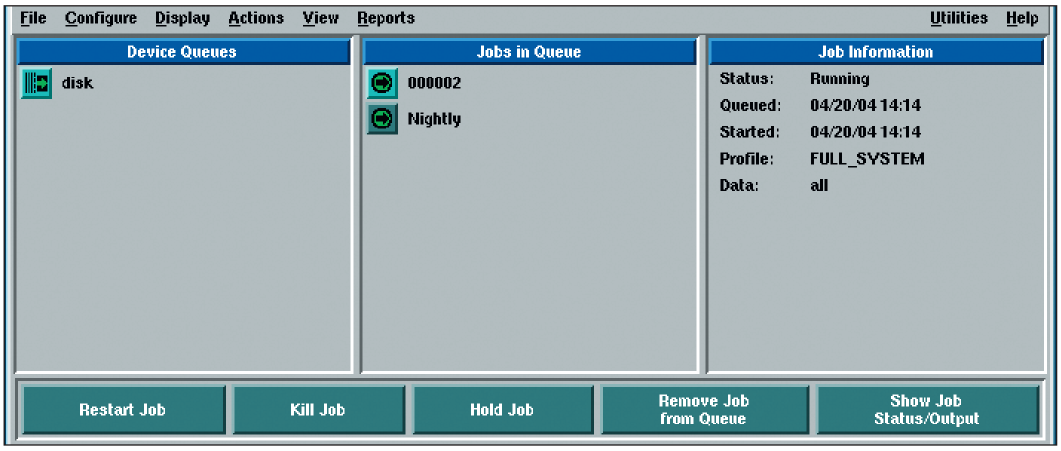
LINUX FORMAT VERDICT

FEATURES	9/10
PERFORMANCE	9/10
EASE OF USE	7/10
VALUE FOR MONEY	10/10

Does backup and restore to a tee, but needs an interface refresh. If you can't part with the cash, at least get the excellent *Personal* edition.

RATING 9/10

●●●●●●●●●○



disk

000002
Nightly

Status: Running
Queued: 04/20/04 14:14
Started: 04/20/04 14:14
Profile: FULL_SYSTEM
Data: all

It's not a complex GUI, but simplicity doesn't necessarily make it any easier on the eye...

Official Fedora Companion

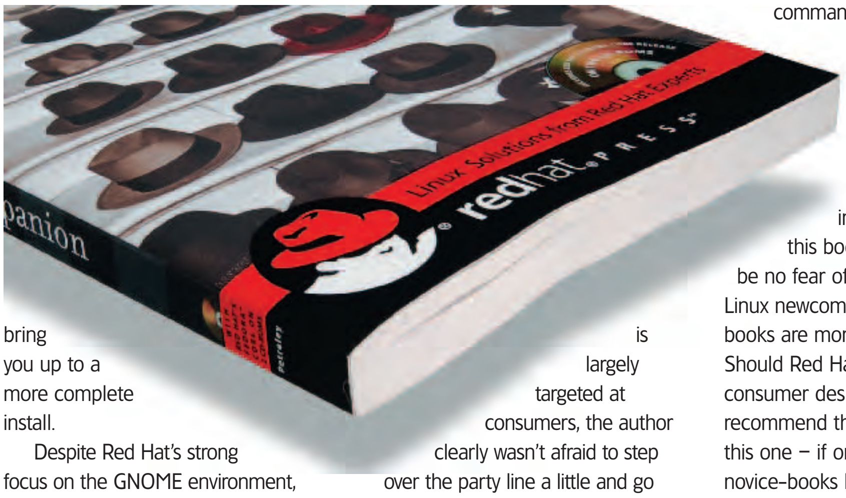
Andy Hudson finds proof that you don't need 1000 pages to teach a distro.

BUYER INFO

- **AUTHOR** Nick Petreley
- **PUBLISHER** Red Hat Press/Wiley
- **ISBN** 0-7645-5836-6
- **PRICE** £13.99
- **PAGES** 260

This book is something of a enigma: after all, how often do you see a book on a distro that has only 260 pages? Clearly that's not enough room to go into depth on every topic, so you may well wonder what the author chose – this is the *Official* guide, remember.

Well, quite a bit it seems. Petreley manages to cover a wide enough range of topics to get/*anyone* started using Fedora Core 1 (FC1). The publishers' version of FC1 is included on two CDs, which is enough to get a fairly solid system up and running. Once installed, you can of course use *YUM* (the package manager) to install any further applications required to bring you up to a more complete install.



Despite Red Hat's strong focus on the GNOME environment, the book doesn't shy away from discussing alternative desktop environments. Granted, Bluecurve removes many of the aestethic differences that would otherwise be apparent between the two desktops, but where differences arise, Petreley does a good job of explaining them. Although the book, like the distro itself, is largely targeted at consumers, the author clearly wasn't afraid to step over the party line a little and go where Red Hat daren't tread – it does, for example, give easy-to-follow instructions for how to enable MP3 support, while also explaining Red Hat's reasons for disabling it. Beyond the straight consumer information that is the core of any book pitched at this level, we were pleasantly surprised to see quite a few pages devoted to shell commands, Unix permissions, as well as some traditional admin tools - this kind of content is becoming increasingly rare in modern 'newbie' books.

Perhaps the most important thing about this book is that there would be no fear of giving it to a complete Linux newcomer, whereas weightier books are more likely to inspire terror. Should Red Hat ever return to the consumer desktop market, we would recommend that it bundles a book like this one – if only there were more novice-books like it in the meantime!

LINUX FORMAT VERDICT

Got friends or family that you want to gently introduce to Linux? Here's the book to help you. Then, when you shout *"RTFM!"* at them – they'll read this!

RATING 10/10

●●●●●●●●●●

Dancing Barefoot

Paul Hudson departs from technical tomes with some masterly heart-cockle warmery...

BUYER INFO

- **AUTHOR** Wil Wheaton
- **PUBLISHER** O'Reilly
- **ISBN** 0-596-00674-8
- **PRICE** £6.95
- **PAGES** 115

Before reading this book, there is one thing you need to get quite clear: Wil Wheaton played Wesley Crusher in *Star Trek: The Next Generation*, but he's not just an actor in that popular television show. In fact, it turns out he's not even much of a fan of *Star Trek: Deep Space 9* or *Star Trek: Voyager*, but we still love him anyway because – above all – he's a geek. Yes, he reads Slashdot (even posting about eight times a month), has his own Geek Code signature, uses Linux, writes a weblog and runs his website, enjoys geocaching, and even did his fair share of role-playing gaming.

The text is broken up into five stories – all true – of which at least two are outright sentimental. The others are primarily funny and often make you stop and think (or laugh!), but, once you're finished reading it, you realise that *all* of the stories are sentimental – even when he describes what it's like taking part in *Star Trek* conventions, you realise how much he actually cares for the fans, even if the odd one does say how much they hated Wesley! The largest story, *"The Saga of SpongeBob VegasPants"*, takes up more than half the book recounting various events in Wil's *Trek* career, jumping smoothly from his experiences on the *Star Trek Experience* ride, filming on set, meeting Captain Kirk, and walking into Turbolift doors when they didn't open in time.

Wheaton writes in a laid-back, chatty manner and truly is a gifted comic. Although he scores "+5 Funny" regularly with his Slashdot posts, there was a chance that these were all just one-liners and that he'd struggle to keep the pace across the book, but there was none of that – he runs off on tangents about Shatner's toupee and about how much he likes playing with his kids in a light-hearted manner that makes you laugh out loud and keep turning the pages. When he's not funny, he's poignant and sincere, which makes for equally compelling reading.

To sum up, I'll leave you with this quote from the book: *"I can't wait to go out and show these people that I've grown up, become funny, and (most of all) that I'm not Wesley Crusher."*

LINUX FORMAT VERDICT

Short but sweet, a highly recommended addition to anyone's bookshelf – Trekker, Trekkie, geek or otherwise – we can't wait for his next book!

RATING 10/10

●●●●●●●●●●

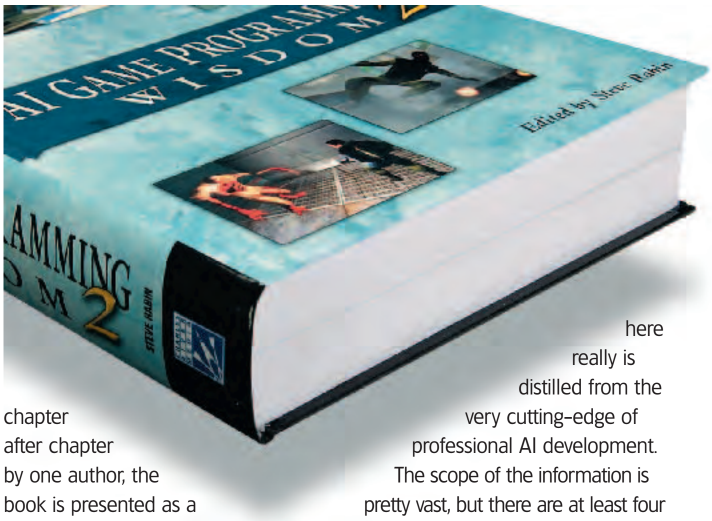
AI Game Programming Wisdom 2

About to embark on an AI voyage of his own, **Paul Hudson** checks his lifejacket.

LINUX FORMAT TOP STUFF AWARD

BUYER INFO

- **AUTHOR** Steve Rabin, et al
- **PUBLISHER** Charles River Media
- **ISBN** 1-58450-289-4
- **PRICE** £46.95
- **PAGES** 732



When it comes to books on games programming, Charles River Media (CRM) is entirely unequalled: its *Games Programming Gems* series (soon to get a fourth member) is on the shelf of every serious game programmer, and the original *AI Game Programming Wisdom* book was both insightful and helpful in a wide range of topics. As we'll be looking at AI in next month's *Trout Wars* tutorial, this book came out at just the right time – but will it help turn your game into a masterpiece?

Wisdom incarnate

The style of the CRM books is what we like the most – rather than having chapter after chapter by one author, the book is presented as a series of essays by individuals who are experts in that one field. This works well because you can jump in wherever you like and learn one technique without having to wade through everything else. This time there are almost 70 essays written by people from all the gaming greats, which means the wisdom presented here really is distilled from the very cutting-edge of professional AI development.

The scope of the information is pretty vast, but there are at least four articles that stand out as real 'must-reads' – *"Artificial Stupidity: the art of intentional mistakes"* discusses why it's better to produce a clever AI that occasionally screws up realistically rather than just program a dumb AI, *"Hunting Down the Player in a Convincing Manner"* talks about techniques to make your AI agents work in a more human-like manner, *"Random Map Generation for Strategy Games"* is an excellent introduction to automatic map creation for RTS games, and *"Transport Unit AI for Strategy Games"*, which is a real gem of an essay discussing the age-old problem of how do you get your AI to build ships to help land-based units cross water – think about it – it's not so easy!

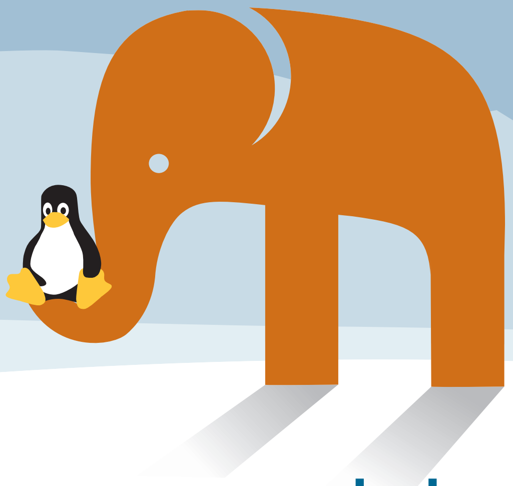
Yet again, CRM has outdone itself with a first-class title that deserves a place on every games programmer's bookshelf. If you already own the first Wisdom book, buying this one is a no-brain decision you've probably already made. If you don't own the first, buy them both and rock the gaming world!

LINUX FORMAT VERDICT

Whets our appetite for *Games Programming Gems 4* like nothing else could – kudos to Rabin for such an excellent production.

RATING **10/10**

●●●●●●●●●●

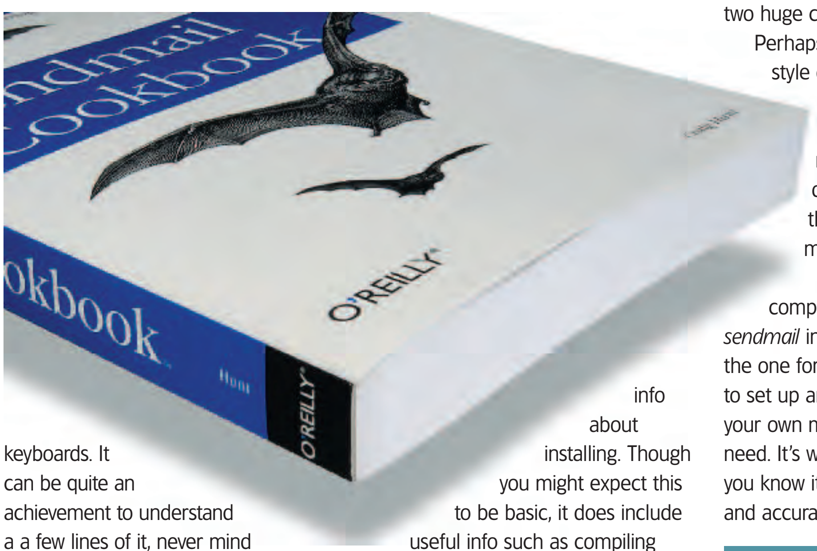


Sendmail Cookbook

Eye of newt and wing of bat, make this spam not come back... **chants Nick Veitch.**

BUYER INFO

- **AUTHOR** Craig Hunt
- **PUBLISHER** O Reilly
- **ISBN** 0-596-00471-0
- **PRICE** £31.95
- **PAGES** 388



S*endmail* is one of those pieces of software that has been around so long, it seems to have always been there. And although there are many alternative MTAs (Mail Transport Agents) for the Unix platform, *sendmail* was the first to achieve some sort of pre-eminence, and is still, as far as anyone can tell, the most popular.

One of the big disadvantages of *sendmail*, as anyone who has any amount of experience with it will tell you, is that while it is undoubtedly powerful, it is also a cumbersome beast. The configuration file looks like it was written in some bizarre language created by a race of people who only had the number row available on their keyboards. It can be quite an achievement to understand a a few lines of it, never mind deduce how you might want to change it. This book then, should be a godsend to anyone who finds themselves unenviably charged with doing just that.

Although this is billed as a *Cookbook* – rather than a *Complete Guide* – it does start off with an introduction to *sendmail*, and the usual info about installing. Though you might expect this to be basic, it does include useful info such as compiling *sendmail* to use LDAP, SASL and various other options.

The real cooking comes in the later chapters. After the usual info about masquerading, and routing sections, you get to the good stuff like automatically rejecting spam, using *Auth* as an authentication method and two huge chapters on security.

Perhaps this isn't in the typical style of the normal O'Reilly cookbooks, in that it deals in detail with a fairly small number of topics, but it does expand and extend on the information available in more general volumes.

If you are looking for a companion to a mission-critical *sendmail* install, the 'other' bat book is the one for you. If you want to be able to set up and do a few cool things with your own mailserver, this is all you will need. It's written by Craig Hunt too, so you know it's going to be easy to read and accurate.

LINUX FORMAT VERDICT

Well written, concise but with plenty of detail in areas that matter, despite it being billed as a *Cookbook* rather than a *Complete Guide*.

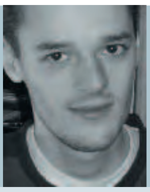
RATING 8/10

●●●●●●●●○○

HotPicks

The best new Open Source software on the planet!

On the CD and DVD



Mike Saunders

A coder since Amiga times, Mike's a Linux and BSD guru.

This is the place where we get to profile some of the hottest software around. Each month we trawl through the hundreds of Open Source projects which are released or updated, and select the newest, most inventive and best for your perusal. Most of the *Hot Picks* are available on our coverdiscs, but we've provided web links if you want to make sure you have the very latest version.

If you have any suggestions for software that you think we should cover, email us at linuxformat@futurenet.co.uk with "Hot Picks Request" as the subject-line, or contact us by post through the address on the *Mailserver* pages.

HOTPICKS AT A GLANCE

HOTPICKS AWARD

HOTTEST PICK!

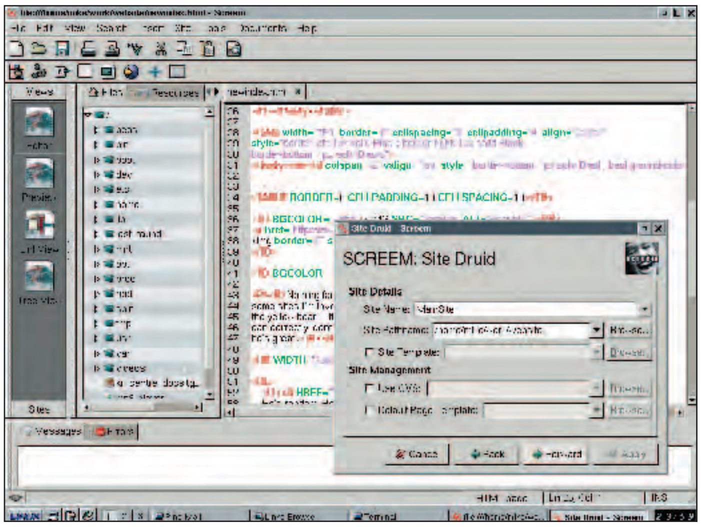
Everything covered in our Hot Picks section is unmissable, but every month we'll be singling out one project for outstanding brilliance. Only the very best will be chosen!

HTML EDITOR

SCREEM

HOTTEST PICK!

■ **VERSION** 0.10.2 ■ **WEB** www.screem.org/



SCREEM: Site Druid

Site Details

Site Name: NewSite

Site Management

***SCREEM* at work on a website, with a Wizard-esque Druid popped up.**

***Emacs*, *Vim* and even *Pico* are all** great for creating basic personal websites, but for larger projects their limitations become apparent rather quickly. Fortunately, Linux is getting ever stronger on the HTML editing front – we have WYSIWYG apps in the form of *Mozilla Composer* and Linspire's *Nvu*, along with more advanced tools such as *Bluefish* and *Quanta*. *SCREEM* (*Site CReation and Editing EnvironMent*) has been around for some time, and as we last looked at it over two years ago, we decided to give it some fresh coverage.

SCREEM is a GNOME app, and consequently requires the relevant development packages installed to compile from source. The only unusual dependency is *libcroco* and we're supplied everything you need on the coverdisc. Various distros have binary packages available; the one on our disc is for Mandrake but should work on other up-to-date installations too.

After firing up an uninspiring splash screen, *SCREEM* presents the main window and – as we're seeing more often with polished Linux desktop apps – a 'Tip of the Day' box giving quick pointers on usage. Those running lower resolutions (*eg* 800x600) may find the default layout to be overly cramped, but most panes and boxes can be shifted around and resized to the best fit. Crisp icons and tooltipped buttons abound, and overall the UI design is sane and easy to navigate.

SCREEM boasts a smart Views panel for switching between different perspectives on the current file. Along with the main editor, there's a coloured links diagram (highlighting local, external and broken links), a tree view of the document's elements and a miniature HTML rendering engine for immediately viewing alterations. This is a tad weak in places and is no substitute for testing the page in *Gecko* or KHTML, but works fine for spotting errors and doing quick checks.

Life's a SCREEM

The editor component is a solid affair with the expected cut/paste, undo/redo and find/replace (over multiple files) functionality all rolled in. It numbers lines and sports syntax highlighting to make working with complex HTML much more pleasant, and there's even a macro facility for assigning strings of text to short keystrokes. When the user starts typing a tag, *SCREEM* offers a pop-up list of possible attributes – it's a handy touch, particularly for intermediate HTML coders, as is the 'Intelliclose' (hah!) feature which automagically fills in the next tag to be closed.

This featureset would be more than adequate for a simple editor, but the Page and Site Druids are where *SCREEM* really comes to life. Akin to the Wizard-type dialogs ever-present on Windows, the Druids step through different phases of site creation: file locations, where and how to upload, text and link colours and other details, finally producing the HTML skeleton to work in. Together with the template system they're helpful little time-savers.

In terms of customisation, *SCREEM*'s Preferences window includes a handful of tweakable settings, such as the default font, disabling certain features in the editor, external browsers and more. The (occasionally outdated) documentation is detailed in some places and quite bare in others – as it's not a 1.0 release yet, though, this isn't a major issue.

Undoubtedly, *SCREEM*'s main competition in the Linux HTML editing arena is *Quanta*, the gigantically popular KDE-based app – there's not a massive amount between them, but *Quanta* just pips it to the post in terms of more advanced features and options. This works both ways though, and *SCREEM* is certainly more accessible for smaller sites and less-experienced HTML hackers.

Another area in which *Quanta* has the lead is stability; we ran into a couple of reproducible crashes in *SCREEM*, although for the most part, in general editing and related tasks, it was well-behaved. On the whole, we're pleased to see *SCREEM* making steady progress and recommend it to anyone looking for a tidy and well-executed HTML editor with a decent range of goodies.

FLOPPY RESCUE DISTRO

SlimLinux

■ **VERSION** 0.7.0 ■ **WEB** http://slimlinux.freezope.org/

New PCs are typically shipped with 256+ MB of RAM, hard drives are well into several hundred gigabytes and DVD writers are becoming ever more commonplace, and yet the good old floppy disk still shows no signs of dying off. Many PCs still include them – they're not much use for data transfer, but when everything falls over they can be a godsend. Every good admin keeps around a CD or floppy-based micro Linux system for rescue purposes, and a recent entry into this bustling field is SlimLinux.

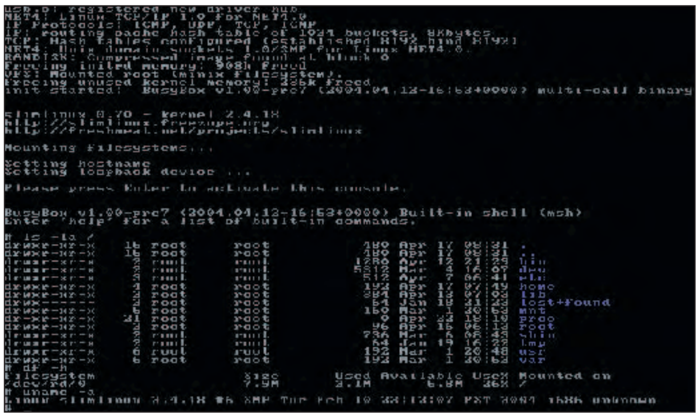
SlimLinux works along the same lines as TomsRootBoot and other rescue floppy distros. The main image file is slightly larger than standard floppy sizes, so you'll need to write it to a blank disk as follows:

```
dd if=slimlinux-0.7.0.img bs=1722k
count=1 of=/dev/fd0u1722
```

This produces a bootable floppy (under Windows, rawrite2.exe should do the trick), and a hard drive alternative for starting from DOS is also available. Depending on hardware, SlimLinux boots with the framebuffer enabled, allowing for much more screen space than the standard 80x25 characters. It dives straight into a root prompt (no password set), and right away the user can start working – it's all lightning fast as it's running completely from RAM.

To conserve space, the SlimLinux distribution makes use of *uClibc* and *BusyBox*, rather than the more common but weightier *glibc* and *Bash*. *BusyBox* is more tolerable than some other minimalist shells though; it has tab-completion and a good range of inbuilt commands. The 2.4.18 kernel is supported by a useful collection of small tools, including the *Emacs*-like editor *Zile*, *Retawq* web browser, *Mutt* mailer and *Clex* file manager. Although these aren't the most featureful applications in existence, they're more than enough to hack config files and pull stuff from the Net.

In many cases, rescue floppies are used to mount the filesystems of broken OS installations, and SlimLinux does OK on this front – ext2, msdos/vfat and iso9660 (CD) are supported in the kernel. The developer has managed to cram an impressive array of utilities into a really small space in SlimLinux, and it's well worth keeping a copy next to your computer in case of emergency.



```
usb.c: registered new driver hub
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 1024 buckets, 8Kbytes
TCP: Hash tables configured (established 8192 bind 8192)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 908k freed
VFS: Mounted root (minix filesystem).
Freeing unused kernel memory: 236k freed
init started:  BusyBox v1.00-pre7 (2004.04.12-16:53+0000) multi-call binary

slimlinux 0.70 - kernel 2.4.18
http://slimlinux.freezope.org
http://freshmeat.net/projects/slimlinux

Mounting filesystems...

Setting hostname
Setting loopback device ...

Please press Enter to activate this console.

BusyBox v1.00-pre7 (2004.04.12-16:53+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

# ls -la /
drwxr-xr-x   16 root     root          480 Apr 17 08:31 .
drwxr-xr-x   16 root     root          480 Apr 17 08:31 ..
drwxr-xr-x    2 root     root         1280 Apr 12 21:29 bin
drwxr-xr-x    3 root     root         5312 Mar  4 16:07 dev
drwxr-xr-x    3 root     root          512 Apr  7 06:41 etc
drwxr-xr-x    4 root     root          192 Apr 17 07:49 home
drwxr-xr-x    2 root     root          384 Apr 13 07:03 lib
drwx------    2 root     root           64 Jan 18 21:22 lost+found
drwxr-xr-x    6 root     root          160 Mar  1 20:52 mnt
dr-xr-xr-x   21 root     root            0 Apr 22 18:10 proc
drwxr-xr-x    2 root     root           96 Apr 16 06:12 root
drwxr-xr-x    2 root     root          736 Mar  6 08:43 sbin
drwxr-xr-x    2 root     root           64 Jan 19 16:22 tmp
drwxr-xr-x    6 root     root          192 Mar  1 20:48 usr
drwxr-xr-x    6 root     root          192 Mar  1 20:52 var
# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/rd/0                 7.9M      2.1M      6.8M  26% /
# uname -a
Linux slimlinux 2.4.18 #6 SMP Tue Feb 10 23:13:07 PST 2004 i686 unknown
#
```

Nothing visually exciting about SlimLinux – just familiar console antics.

FILESYSTEM EXPLORER

XCruiser

■ **VERSION** 0.30 ■ **WEB** http://xcruiser.sourceforge.net/

3D graphical interfaces are all the rage at the moment – Sun's spiffy *Looking Glass* has generated plenty of discussion around the Net, and various other projects for 3D window managers are underway. Similarly, we've seen the outrageously cool *psDooM* appear (play *Doom* to kill enemies representing processes) and some efforts to 3D-ify file navigation. *XCruiser* recreates filesystems as 'universes' to explore, and was suggested for inclusion this month by *LXF* reader Dan Burton.

XCruiser is available as small binary and source packages; we've supplied both on our coverdisc, and the RPM should work on almost all recent distros. Failing that, building from source should be straightforward – you'll need to enter

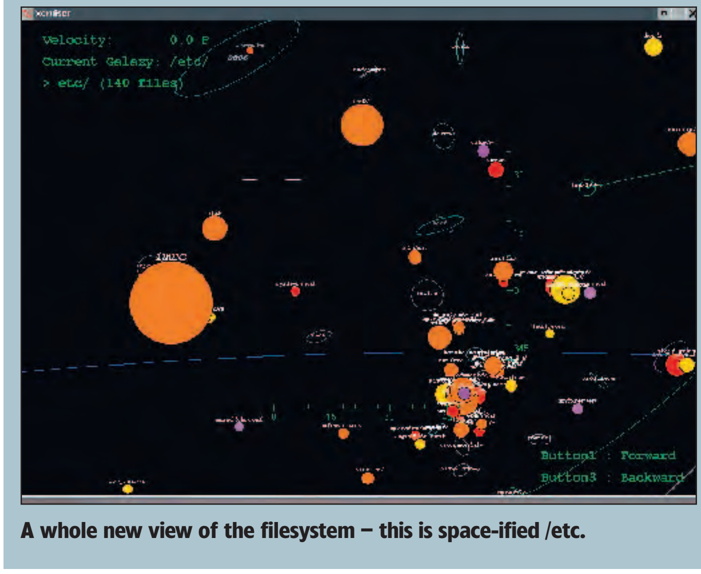
```
xmkmf -a
make
make install
```

(as root) to put everything in place. Once done, entering **xcruiser** at a terminal prompt will fire it up.

Cruising around inside the virtualised filesystem involves pointing the mouse and holding the left and right buttons to move forwards and backwards respectively. Directories are depicted as outlined circles in which file 'planets' lie, their sizes relative to the files themselves, and symbolic links can be seen as green arcs. As it's a simplistic 3D environment, it runs smoothly even on fairly low-spec boxes.

Traditional file managers are generally limited to viewing a handful of directories at a time, but *XCruiser* provides a unique – and often spectacular – perspective, with the ability to spot distant directories and gauge how busy they are. It's bizarrely entertaining, and entertainingly bizarre. Moreover, it suddenly hits home just how many files there are on a typical Linux installation nowadays – flying over to /usr/share/doc may well get the hard drive grinding.

At present, *XCruiser* has a hard-coded limit of 256 celestial objects for each directory, so a journey into /dev won't leave you bewildered in a sea of device nodes. We'd love to see some kind of weapon for deleting files, or even a cargo scoop to, er, move them to other directories, although the potential for damage could be disastrous. Nonetheless, it's a fun little application – if you can find a genuine use for it, do let us know...



A whole new view of the filesystem – this is space-ified /etc.

SMB WEB INTERFACE

SmbWebClient

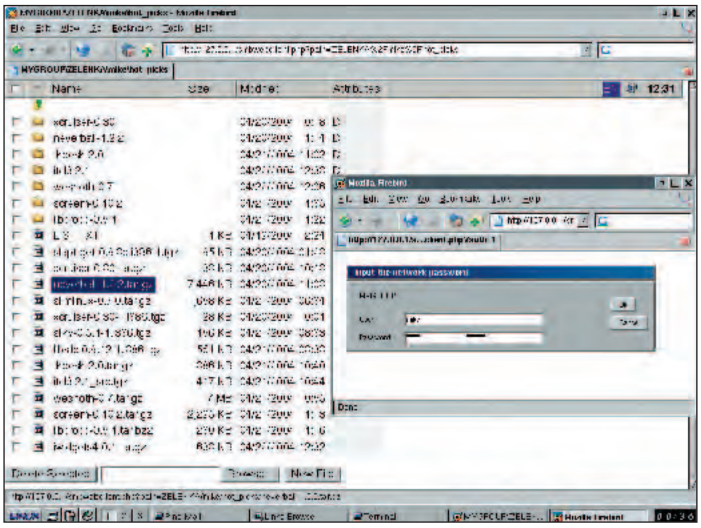
■ **VERSION** 1.78 ■ **WEB** www.nivel0.net/SmbWebClient/

Linux has been making great inroads into the small departmental server space thanks to *Samba*, the highly acclaimed interoperability suite for replacing Windows file and print servers with Open Source equivalents. On the client side, *Nautilus, Konqueror* and XFce's *XFFM* (among others) can browse SMB shares, and now *SmbWebClient* presents an intriguing platform-independent alternative.

Setting up *SmbWebClient* is a breeze – it's a single standalone PHP script, requiring only PHP 4.1 (or later) and the *smbclient* binary as supplied with *Samba*. Before copying the file into an appropriate place on your webserver, you'll need to edit the first few lines of it – these contain general configuration bits and bobs, and thankfully it's all pretty well commented.

If everything's correct, you should be able to access the script via your browser and view the SMB machines on the network; if this fails, try altering the configuration and possibly pick out the problem from the logfile (specified with **cfgLogFile** in the script). *SmbWebClient*'s log format is virtually identical to *Apache*'s, making it usable in many of the analysis tools floating around the Net.

Simple tables and a sprinkling of JavaScript are used to build *SmbWebClient*'s pages, and we came across no problems when testing it with *Gecko* and KHTML. It's a frill-free and approachable design, with the ability to sort the list of files by name, size or date and select all files in the current directory with one checkbox. *SmbWebClient* includes a clock at the top-right of each page (no auto-refresh, so it's not always accurate) along with a button allowing the user to enter his/her password.



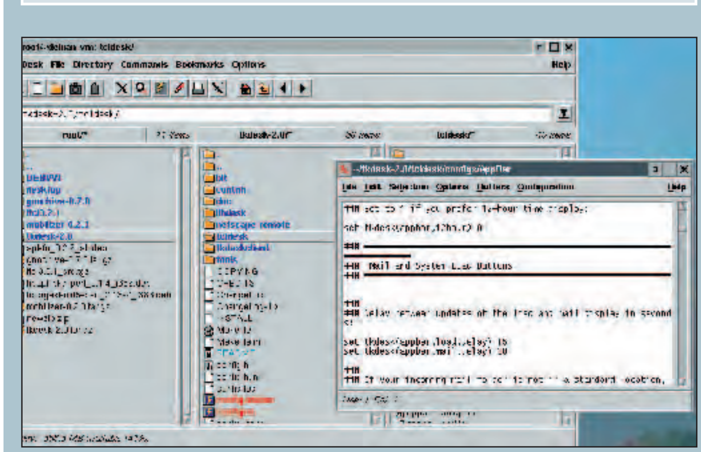
Two *Firebird* windows, showing the file browser and password input box.

Clicking on a file link displays it in the browser window, and files can be deleted and uploaded too – anyone with basic file manager familiarity should have no difficulties getting to grips with it. The Windows-style icon set and general running speed are plus points too. As such, *SmbWebClient* is a quick and no-nonsense solution when full SMB clients are overkill for your users, or on more exotic desktop OSes that don't have native clients.

FILE MANAGER/DESKTOP

TkDesk

■ **VERSION** 2.0 ■ **WEB** http://tkdesk.sourceforge.net/



***TkDesk* in action, with the file manager, appbar and editor up front.**

One of Linux's great strengths is its facility to be able to make good use of older hardware. KDE and GNOME may be heavy beasts and require fairly recent machines to run well, but there's a great deal of lightweight tools and apps out there as well – *TkDesk* is one, and *LXF* reader Ruadh O'Conachar asked us to give it a bit of coverage.

If you're up for some horrendously complicated installation work, you won't go wrong with *TkDesk* – it uses a mixture of Tcl/Tk, C and the IncrTcl addition, and can be a nightmare to build. It segfaulted on startup on our Slack 9.1 test box, wouldn't build properly on Mandrake 10.0, and only just worked on Debian 3.0 (you may have to edit each subdirectory's Makefile to specify the Tcl and Tk includes, even with the **--with-tcl** flag to **./configure**). Egad.

Initial complications aside, once *TkDesk* is running, it zips along at a refreshing pace – orders of magnitude faster than the weightier desktops, and this is one of its main attractions. The Tk widgets are chunky and solid, and they don't look too out of place with *Motif* and *GTK1* (default theme) apps.

TkDesk's two main components are the file manager and the application bar (appbar). Together with an appropriate window manager, these provide the basics of a desktop environment; the default appbar setup contains icon links to various programs, while the file manager uses the usual window furniture (toolbar, status line *etc*) and context menus.

Some of the file manager's notable features include a bookmarks system, drag-and-drop (middle mouse button), file searching and a primitive text editor. Much of *TkDesk*'s configuration is done with this editor – a few menu option toggles are available, but the commented config files allow for greater flexibility.

It's a shame that *TkDesk* is rather a chore to get up and working on some modern distros, as it's a speedy and slender alternative to *Konqueror* and *Nautilus*, ideally suited to older boxes with smaller window managers such as *FVWM* or *Fluxbox*. If you're looking for a versatile desktop which doesn't eat up system resources like there's no tomorrow, give it a try.

SILC CLIENT

Silky

■ **VERSION** 0.5.1 ■ **WEB** http://silky.sourceforge.net/

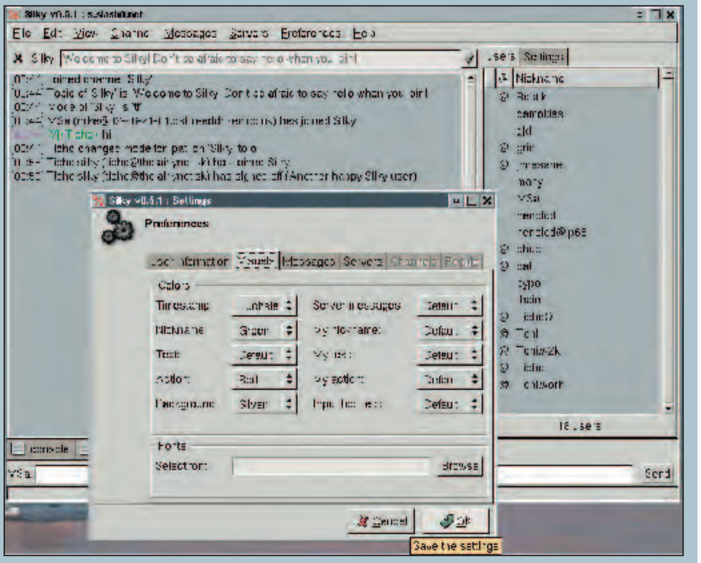
Internet Relay Chat is one of the most popular means of online communication – it's free, easy to use and fast. Unlike Web-based chat rooms which can munch through bandwidth and limit the user to his/her browser, IRC is entirely open and many excellent Linux clients exist. One major snag, though: IRC is not inherently secure. The SILC (Secure Internet Live Conferencing) protocol was developed to remedy this, and *Silky* is a SILC client.

Kudos to the *Silky* developers for producing a single, all-encompassing RPM – it's been tested on RHEL, Fedora, Mandrake and SUSE, and should work on other distros too (no hassles on our Slackware box). As well as the main package you'll need *libsilc*, and if you're compiling from source you'll need *GTK2* development headers and extras installed too.

When first started, *Silky* generates its crypto keys (either automatically or with manual input), and once finished it's ready to start chatting. Users of *X-Chat* and similar IRC clients will feel completely at home with the interface; tabs are used for individual channels (a great screen real-estate saver), the nick list is shown on the right, and menu items are all self-explanatory.

Furthermore, general SILC commands are identical to their IRC counterparts, so /SERVER, /JOIN and /WHOIS etc. all work as expected. One notable feature of the SILC protocol is MIME messages, such as images, and *Silky* includes some functionality for handling these. There's not a great deal in the way of preferences – the basic colour scheme can be modified, as can the server/channel lists and quit message, but that's about it. The documentation is minuscule too, and a short guide to the regular commands would be a bonus for newcomers (although most common operations can be achieved via the menus).

Because IRC has many positives, as mentioned above, it's great for instant communication, and SILC supplies a solution to the big problem of security. Everything along the wire is encrypted and authenticated, and *Silky* is a small, clean and stable client – not world-beating on the snazzy feature front, but very pleasant all the same.



Silky's interface won't be at all alien to regular IRC users.

PACKAGE MANAGEMENT

Slapt-get

■ **VERSION** 0.9.8c ■ **WEB** http://software.jaos.org/

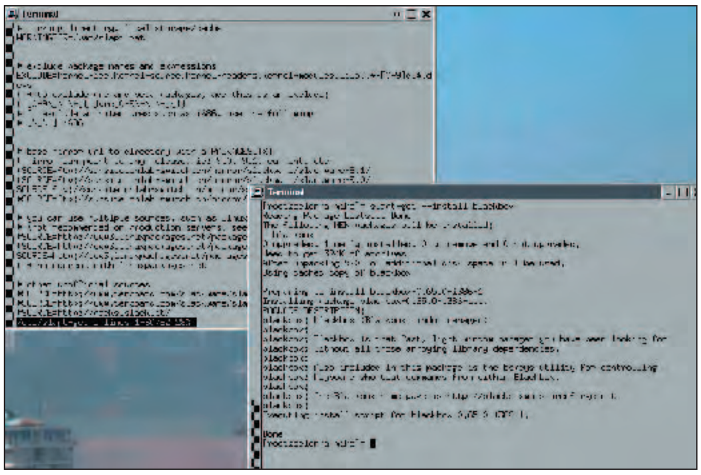
One of the many reasons the Debian distribution has been so highly regarded throughout Linux's history is its Advanced Package Tool, or apt. This incredibly flexible and powerful package handling utility gave Debian the edge for many years; thankfully, though, much of its functionality has been ported or rolled-in to other tools. Meanwhile, Slackware's package management has always taken the small-and-simple approach, and *Slapt-get* strives to bridge the gap.

Slapt-get is supplied as standard Slackware .tgz packages for the last three major releases, and is installable via the usual **installpkg** command (8.1 users have a statically linked version to work around some problems). It relies on the *curl* utility to pull files from the relevant servers – this should be installed by default or can be found in the usual Slackware archives.

Although *Slapt-get* is pretty much set up from the get-go to download packages from a main Slackware archive mirror, the config file in /etc/slapt-getrc allows for more thorough tuning. Certain packages can be excluded from installation or upgrade via names or regular expressions, and extra sources can be added either as URLs or or filesystem locations. From there, a quick **slapt-get --update** brings the package list up to date, and **slapt-get --install <app>** will fetch a program and install it.

All pleasingly simple, then, but *Slapt-get* has plenty of other tricks up its sleeve too. Package information can be retrieved with the **--show** flag, while rudimentary dependency handling is available for those packages which support it (a feature notably missing in Slackware's supplied tools). Upgrading individual packages, patches and entire releases is catered for too, and on the whole it shouldn't be too difficult for users of *apt* on other distros to fathom out – the superbly detailed FAQ helps out here.

Slackware's simplicity and stability are highly attractive to experienced Linux fans. *Slapt-get* (and similar tools such as *Swaret*) brings much simpler and quicker package management to this ever-popular distro, and as a result it's supremely useful for newcomers, power-users and server admins alike. Wonderful stuff.



```
# exclude package names and expressions
```

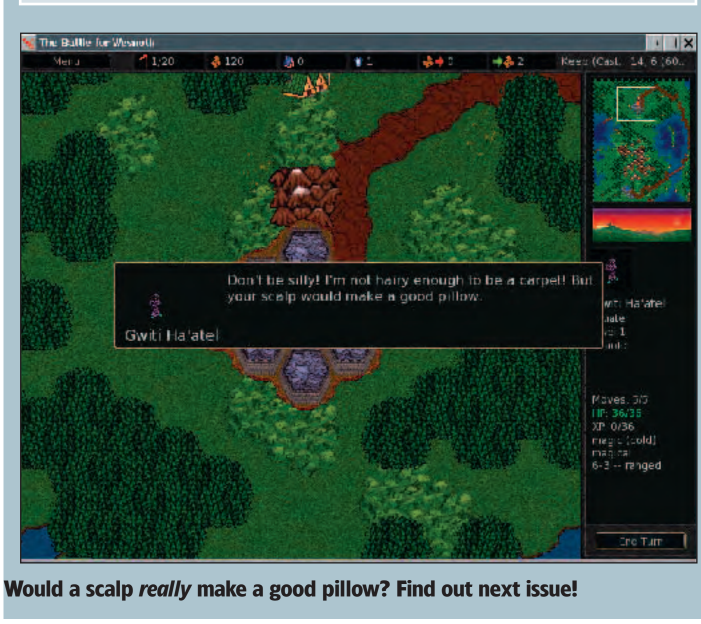
```
slapt-get --install blackbox
Done
```

Top-left: the config file. Bottom-right: *Slapt-get*ting the *Blackbox* WM.

STRATEGY GAME

Battle for Wesnoth

■ **VERSION** 0.7 ■ **WEB** www.wesnoth.org



Don't be silly! I'm not hairy enough to be a carpet! But your scalp would make a good pillow.

Gwiti Ha'atel

Would a scalp *really* make a good pillow? Find out next issue!

Adventure and strategy games are ten-a-penny in the PC world, with all manner of fantasy worlds and tremendous battles being brought to life via commercial titles and now open source alternatives. *Battle for Wesnoth* is one of the best-known turn-based strategy games for Linux – it has a busy surrounding community of developers, translators and gamers, and is highly regarded among long-time strategy and role-playing fans.

Compiling *Battle for Wesnoth* from source shouldn't pose any major problems providing you have SDL and some extras installed (image, mixer, ttf and net). We've included an RPM package and a statically linked binary set on our coverdisc. If you're presented with constant segfaults when trying to start *Wesnoth*, try switching to a 16-bit colour depth and 1024x768 resolution.

On the first go, it's best to become acquainted with the gameplay mechanics via the built-in tutorial mode; this explains the basics of the story behind the game and then goes on to illustrate moves and tactics. As is the norm with a broad slew of strategy titles, your job is to manage a gang of warriors, taking land by force and improving your experience levels along the way. You're given a number of scenarios to work through, each containing objectives to accomplish, and the game's plot unfolds as you progress.

Fighting CPU opposition over the three difficulty levels is all very well, but strategy games are far more exciting when facing real flesh-and-blood enemies, and *Wesnoth* includes an online multiplayer mode (willing players can be found on an IRC channel). And despite the complexities inherent in deep tactical games, the developers have produced some excellent documentation in the form of Getting Started and reference guides.

Strategy games rarely get the adrenaline pumping, but *Battle for Wesnoth* is boosted by lavish visuals and engrossing gameplay – there's plenty for fans of tactical battle epics to sink their teeth into. It's not the kind of game you can pop up for a quick five-minute diversion, but for something rather more cerebral than the usual action fare, it's definitely worth investigating.

ACTION PUZZLE GAME

Neverball

■ **VERSION** 1.2.2 ■ **WEB** http://icculus.org/neverball/

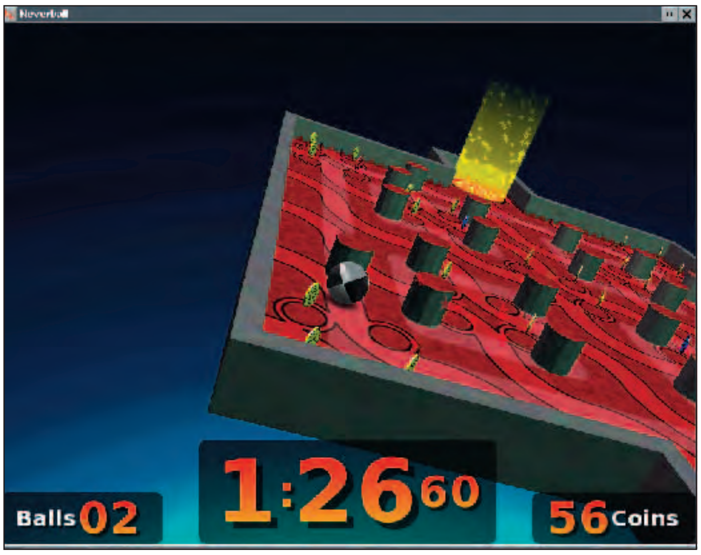
Where would we be without spheres? It'd bring a new lease of life to football for starters, and volleyball... well, that'd be about as playable as *Dark Castle* on the CD-i. Ball-based video games, particularly puzzlers, tend to offer plenty of scope for game designers, and Neverball is no exception. Inspired by *Marble Madness* and Sega's *Super Monkey Ball*, this landscape tilting time-eater was recommended this month by regular *LXF* reader Dave Wickham.

Currently, the only available Linux binaries are somewhat outdated, so it's best to compile from source. This shouldn't require any big stunts or hacks – just extract the archive, type **make**, and then run it in place with **./neverball**. You'll need SDL (and the associated *mixer, image* and *ttf* add-ons) and if you encounter a -lGLU build problem, uncomment **X11_PATH** in the Makefile.

Neverball is crackingly well presented: smooth AA fonts, translucency, gentle textures and apt background music all join up to great effect. The in-game BGM, always tricky to get right in puzzlers, is muzak *par excellence* here – cheerful tunes alternate with lively tracks (depending on level) and work perfectly. Happily, the game's 3D engine runs zippily, even on relatively old boxes too.

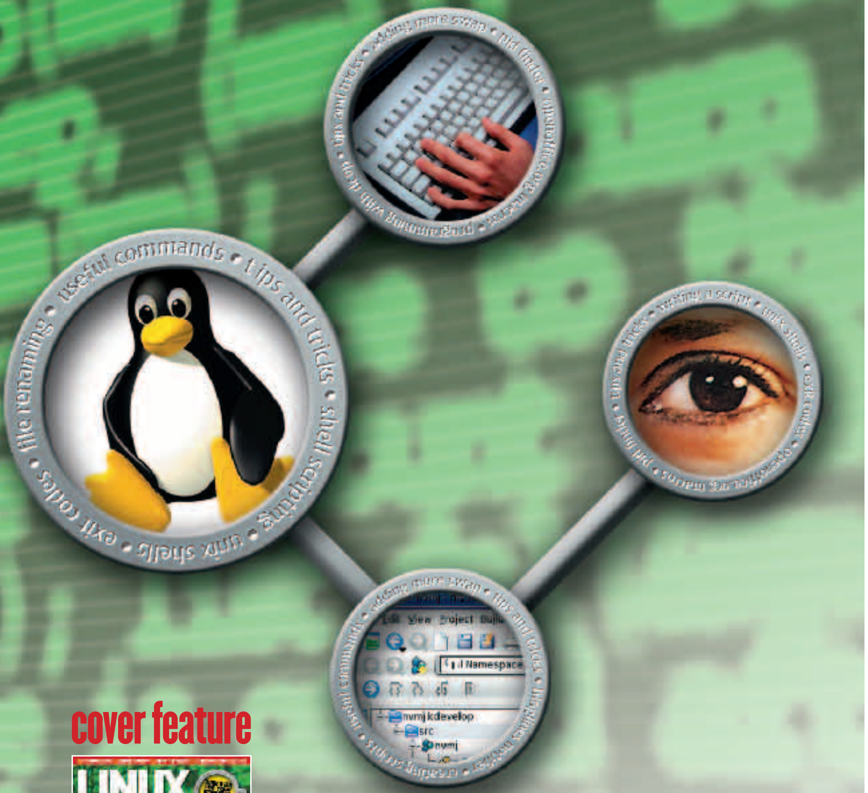
Over a vast number of taxing levels, *Neverball*'s main game goal is to shift the underlying landscape and roll a ball towards the finish. All pretty straightforward? Well, it would be, if it wasn't for all the obstacles *en route* – barriers, holes, moving columns and other irritations all stand in your way, governing the difference between miserable failure and uplifting glory. Coins can be collected on the way as bonuses as well. Although there are three camera angle choices, the game can become a tad frustrating on lengthy levels – the camera tends to move to awkward positions. Still, this just mandates maximum precision with the mouse.

A small sub-game based on *Neverball*'s engine is also included; it's a hugely enjoyable and effective multiplayer crazy golf romp with some stupendously mad courses. All things considered, *Neverball* is a compelling little puzzler with clever challenges, satisfying gameplay and it's marvellously polished to boot. Lovely. LXF



Motion sickness aplenty on one of *Neverball*'s tougher courses. Bleurgh!

Learn to love CODE


useful commands • tips and tricks • shell scripting • unix shells • exit codes • file renaming •

cover feature

Scared of writing programs? Don't be! Straightforward scripts are easier than you think – make your computer work for you, rather than the other way around...

Along with setting the VCR and understanding the opposite sex, programming remains a black art for many people. Ten years ago, this was perhaps justified: geeks were perceived to be people whose shirt pockets were full of pens, had sticky tape holding their thick glasses together and shuffled to *Star Trek* conventions wearing sandals. But in the 21st century, where owning an MP3 player has morphed from the preserve of the technologically hip to the commonplace, broadband Internet access is the norm and nearly everyone went to see *Lord of the Rings* at the cinema, it's clear that due to today's tech-integrated lifestyles, *everyone* is now a geek to some degree.

As Linux users, we're fortunate that our chosen OS comes bundled with all the tools we could need to make it jump to our commands – there are dozens of different ways you can program a Linux box, many of which don't even require you to be a hardcore coder. From shell scripting to automate everyday tasks, to writing *OpenOffice.org* macros to help smooth your workflow, everyone has something to gain from learning at least a little about coding.

Why program?

What would you expect to hear if you could ask Mozart why he wrote *The Marriage of Figaro*? What about if you asked Michelangelo why he painted the ceiling of the Sistine Chapel? Both of these two greats were creative geniuses that simply wanted to produce work that pleased themselves and pleased others – and this is no different from programmers. If you're thinking that developers derive little pleasure from what is essentially shifting little 1s and 0s around a hard disk, you'd be wrong – many people program for fun. Those who do will usually cite one of the following reasons:

1 They like the challenge
Programming is about solving problems in newer, better ways, and there's a huge, open field for people where people can try out new ideas.

2 They like the control
Programming is about logic, as (correctly working) computers only do what they are told to do. Thus, a programmer with a little skill can make a game, make music, or program their PC to solve a *Rubik's Cube* – all with just the tools on their computer.

3 They like to learn new things
Programming is a huge area with many different disciplines – a games programmer might specialise in anything from graphics to audio, or AI to game physics, which means there's a lot of diversity. Add to that the fact that technologies are evolving all the time, which means that even programmers experienced in one field will learn new things in the same area on a regular basis.

4 They like to create things
Programming means you design and

implement an application that solves a problem, and usually (unless there have been some serious problems during development!) the programmers involved are proud of their product. Put simply, there's nothing quite like the buzz that you get when someone emails you to say how much they liked an application that you wrote!

5 They like to work to their level Programming isn't all as hard as KDE or the Linux kernel – there are thousands of problems that can be solved with little or no previous programming experience, which means everyone can bite off a chunk that suits them start toying around at their own pace.

So, with all this in mind, let's take a look at the basics of programming...

Programming ABC

Even if you've some experience in programming in the past, we suggest that you don't skip this section, because here we're going to be looking at what actually makes up a program – the absolute basics of programming – and it'll be a good refresher for you. An outline of programming basics is something that's been requested many times in the past by *LXF* readers, so it's about time we got around to printing it!

The highest level of a program is the program itself, and is often called several different things. *'script,' 'binary,' 'executable,' 'application'* and *'applet'* are all common names for programs, and are pretty much synonymous unless you're hostage to the pedantries of professional development. Each program is created using a programming language, and there are many hundreds of languages available. However, the point of them all is the same: they allow you to control your computer.

One of the main reasons there are different languages is that they each offer different levels of abstraction. A very simple language, such as BASIC, often has one line of code evaluate to thousands of machine code 1s and 0s, which is the only thing computers can work with. Furthermore, these easier languages do more work for you – using BASIC as the example again, it will automatically convert numbers to text and text to numbers, and even automatically stop you from many coding errors. The next question is, *"so why doesn't everyone use BASIC?"* To which the answer is that all the extra work that BASIC does behind the scenes takes up valuable processor time. A BASIC program usually executes about 25-50 per cent as fast as the equivalent C program.

Although there are quite a few languages to choose from, the vast majority of languages share a set few operations. Thanks to this, once you grasp these basics, you should be able to pick up other languages very easily.

These shared operations are: *variables and constants, comments, functions, conditional statements, loops,* and *case switching*. Although they may look different in various languages, they usually work in the same way.

Variables and constants These are types of data store – where you can store information about a piece of data you want to use in your program. This might be the name of the user (this changes, so it's 'variable'), the unique identification number of the hard disk (this doesn't change, so it's 'constant'), or various other things. There's no real limit to the number of variables and constants a program can have.

Variables have data assigned to them at runtime.

So, in C:

```
c = 5;
c = 10;
```

And in Pascal:

```
c := 5;
c := 10;
```

Comments These allow to you insert plain old text into your programs to help you remember what certain parts do. Comments are marked in different ways according to the language you use. In C++, comments start with //, in BASIC they start with **'**, in Pascal you surround them with braces (the **{** and **}** symbols), and more. You only need remember the comments specific to your current language.

Functions If your program ran from the first line to the last line without any way to move around to re-execute certain parts, complex programs would be very long. Functions allow you define a block of code, then run it from elsewhere. This promotes code re-use, and also makes your code less buggy. In PHP, a function looks like this:

```
function myfunc($foo, $bar) {
  print "$foo and $bar were passed
in";
}
```

With that done, you can call **myfunc()** any time you want to, sending with it two bits of data. Note the **$foo** and **$bar** in the function name – these are parameters, and mean that those two variables can be used inside the function.

The function could be called like this:

```
myfunc("hello", "world");
```

That would output "hello and world were passed in", as the **$foo** and **$bar** get replaced with **hello** and **world** when the PHP code runs.

Conditional statements These allow you to execute a block of code only if a certain condition is matched, such as *"if the age variable is greater than 12"*. In C++, a conditional statement looks like this:

```
if (age > 12) {
    output = "You're older than 12";
} else {
    output = "You're younger than
12";
}
```

Loops These allow you to execute a chunk of code several times. Most languages support three basic types of loop, **for**, **while** and **do...while**. Of those, only really the first two are used regularly. Here's an example of a **for** loop in C:

```
for (i = 0; i < 100; i = i + 10) {
  // code here
}
```

And here's the same for loop, this time in PHP:

```
for ($i = 0; $i < 100; $i = $i + 10) {
  // code here
}
```

As you can see, the two are almost identical. Java also looks quite similar to C, which means you can learn three languages in one go! The for loop works by calling the first part (**$i = 0**), then executing the code as many times

»

USEFUL COMMANDS

cd – Change current working directory

pwd – Displays current working directory

ls – Lists contents of current directory

mv – Moves, or renames, a file or group of files

cp – Copies files from one path to another

rm – Deletes files

whoami – Tells you who you are logged in as

ps – Lists current processes on your shell

kill – Sends a signal to a process, usually in order to kill it

grep – Only outputs the parts of the input which match a particular pattern

more – Displays input content a page at a time

man – Displays manual page for the command that it prefaces

info – Displays the info page for a command

w – Lists all users who are logged in and their terminal

wc – Displays the number of characters, words and lines in the input or a file

> **"Programming is a huge area with many different disciplines – even experienced programmers are learning all the time."**

« as it takes until the second part (**$i < 10**) is no longer true. Each time it executes the code, it executes the third part (**$i = $i + 10**) also, which means the first time it goes around, **$i** is **0**, the second time it's **10**, the third it's **20**, etc, until it reaches **100** and fails the test (100 is not less than 100) and stops.

The while loop is similar to the for loop except you need to make it exit yourself. Here it is in C:

```
mynum = 0;
while (mynum < 10) {
  mynum = mynum + 1;
}
```

Case switching This allows you to check a variable against a large number of possibilities quickly and easily. Consider the following situation: you have an Age variable, and you want to print out different text depending on whether the age is less then 10, between 11 and 20, 21 and 30, 31 and 40, 41 and 50, 51 and 60, and 61+. Using standard conditional statements you'd end up with quite a few **if**s, which makes your code look cluttered and not very fast to execute.

Instead, we have case switching, which is essentially a way to check a variable many times. Here's an example in PHP:

```
switch ($name) {
  case "Paul":
    print "Your name is Paul";
    break;
  case "Jack":
    print "Your name is Jack";
    break;
  case "Sam":
    print "Your name is Sam";
    break;
  case "Alan":
    print "Your name is Alan";
    break;
}
```

There what will happen is that the variable **$name** will be checked against **Paul**, **Jack**, **Sam**, and **Alan** (in that order), and, when it finds a match, it will execute the two lines of code beneath it. Now, the **break;** line is special – it means exit the case switch immediately and go to the next line. If **$name** was Jack, the program would print out "Your name is Jack", then, if there was no **break** statement immediately afterwards, it would automatically print out the contents of the **Sam** block and the contents of the **Alan** block.

This is a little counter-intuitive, but there are some times when it's helpful to have this 'fall-through' behaviour, like this (in C):

```
switch (score) {
  case 1:
    output = "The worst!";
    break;
  case 2:
  case 3:
  case 4:
    output = "Very bad";
    break;
  case 5:
  case 6:
    output = "Average"
    break;
  case 7:
  case 8:
  case 9:
    output = "Not bad";
    break;
  case 10:
    output = "The best!";
    break;
}
```

In that code there's nothing for scores **7** and **8**, which means it will fall-through and execute the same code as score **9**. This grouping makes code easier to read and write, which means using the break statement is best used sparingly!

Now that you're in possession of this fairly simple knowledge, you already know more than enough to start programming straight away – good luck, and remember: *programming is fun!*

SHELL SCRIPTING

Writing shell scripts to automate processes is a dying art, but depend on *Linux Format* to explain how it has real-world applications.

With traditions set in the land of the command line, Unix-based platforms have always been a haven for those who want to find the most efficient (and often obscure!) way to do the simplest of tasks. On the other hand, being able to do many things efficiently from the command-line – particularly repetitive tasks – requires an understanding of scripting and understanding how best to articulate a process in a logical format that can be scripted.

One of the most basic ways of scripting is to use *'shell scripts'*, which are interpreted through a shell process, such as *bash, csh,* or *ksh*. Indeed, much of the functionality of Linux systems – including the vast majority of the boot process – is implemented in a variety of shell scripts. The *de facto* shell for scripting on Linux – and indeed most Unix platforms – is bash, though similar principles apply to other shells; particularly *ksh,* from which *bash* derived much of its functionality.

Shell scripts are simply lines of code, which are interpreted on the fly by the shell interpreter. If we type five lines in a sequence to perform a function from the command-line, we can simply create a script that contains those five lines; and by executing the single script, we can perform the same function. All scripts start out by telling the system which binary to start in order to do something with the script. Anyone who has experience of using CGIs on web servers will be used to specifying the location of the perl binary in order to make them execute, and creating a *bash* script that runs through the *sh* binary is identical. If we were writing a *bash* script, we would place the following at the very beginning of the script:

```
#!/bin/sh
```

The file containing the script must be set to executable, so that we can run it from the command-line, or from within a GUI, and have the system know it's a script, rather than having to run it through the specific interpreter it was written for. Usually, scripts will want to be executable by any user, unless there is a specific reason for them to only be readable by one user; for instance, if it contains passwords or is only functional from within a single user account. The standard permissions for a shell script are world read and execute, and owner write. Obviously, the script must be readable by whomever is going to execute it, so it makes little sense to make it world executable and only readable by the owner. This can be set using **chmod** from the command-line:

```
$ chmod 755 script.sh
```

where **755** is the octal value for the permissions we want to set. It is convention to end a shell script with **.sh**, to signify that it is not a binary, though Unix systems do not pay any attention to the extension on a file.

Getting started

The most obvious script to start out with is the traditional ‘Hello World!’, which is rather simple.

```
#!/bin/sh
echo Hello World!
```

Rather than being a built-in command – of which *bash* has a number – we have to call the external binary **echo** in order to write something to output. As **echo** has no comprehension of strings or variables, it will simply output whatever we tell it to. If we want to put ‘Hello World!’ in a variable, the variable will be replaced with its contents by the interpreter, prior to **echo** ever being executed.

```
#!/bin/sh
STR=”Hello World!”
echo $STR
```

Note that when we define a variable **STR**, we have to quote the string, otherwise the interpreter thinks we are trying to pass the argument **World!** to the **STR=Hello** command. *bash* interprets commands in a very simple way: everything is separated by spaces, and everything prior to the first space is considered the command, and each space-delimited string following is a separate argument.

The **mv** command is a very simple example of this. If we want to rename a file, we would do:

```
$ mv old.file new.file
```

In the case of this command, the shell will execute **mv** and pass **old.file** and **new.file** to the binary as arguments, rather than having the executable parse out the command-line itself.

Loops

Performing a looping process in *bash* can be a little obscure, as the way it handles **for** statements is unusual. Rather than specifying two bounds and a step value, we supply the **for** command with all of the values we want it to loop through. If we want a simple **for** loop that goes through one to five, we can do:

```
for i in 1 2 3 4 5; do
echo $i
done
```

Alternatively, rather than writing out all of the separate values, we can use **seq** to generate the list:

```
for i in $(seq 1 5); do
echo $i
done
```

By wrapping the **seq 1 5** command within **$()**, rather than use **seq 1 5** as the values it loops through, *bash* will execute the **seq** command and use the output as the values for the loop. It’s worth remembering that *bash* has no comprehension of a difference between strings or integers, so we can loop it through **1 2 3 dog 4 5 cat** without any problems.

If we have a list of files, we can loop through those too:

```
for f in *
do
   echo $f
done
```

Whenever we do **rm -fr** * on the command-line, **rm** never sees the * wildcard. Instead, the shell will insert whichever files match the wildcards we pass, which is exactly what happens within our **for** loop. This makes it very easy to **bzip2** log files, or to run through a list of files and rename them:

```
for f in ?*.log
do
    N=$f.$(date +%Y%m%d)
    mv $f $N
    bzip2 $N
done
```

We can also write a very short script to rename the extension on specific files:

```
#!/bin/sh
for filename in *.$1
do
  mv $filename ${filename%$1}$2
done
```

Rather than specifying the variables within the script, we take them from the command-line when the script is executed. **$1** is the first argument, **$2** is the second and so forth. When we **mv** the file, we take the extension specified in **$1** off the **$filename** variable, and add **$2** to the end. We would run the script with:

```
$ ./reext.sh jpg jpeg
```

We can also use the **shift** command to step through arguments if we have an unknown number of arguments to deal with, such as files or usernames.

```
until [ -z “$1” ]
do
  echo -n “$1 “
  shift
done

echo
```

shift not only works for arguments passed to a script, but also for arguments passed to functions. Functions are defined much as with any other language, although *bash* requires them to be defined prior to anything that calls them.

```
#!/bin/sh
lazy ()
{
echo Being lazy for $1 seconds....
sleep $1
}

lazy 30

```

SHELL RECIPE:FILE RENAMING

It’s usually fairly common to have to rename a whole directory full of files which have their filenames formatted in an incorrect way. Whether we want to switch around the date, change specific sections of the filename or clean up badly formatted files generated by yet another bad script. The script below allows us to rename specific portions of a file, as described by the **sed** function, so we can modify files without much effort.

```
#!/bin/bash

ARGS=2
E_BADARGS=65
ONE=1   # For getting singular
/plural right (see below).

if [ $# -ne “$ARGS” ]
then
  echo “Usage: `basename $0` old-
pattern new-pattern”
  # As in “rn gif jpg”, which renames
all gif files in working directory to jpg.
  exit $E_BADARGS
fi

number=0   # Keeps track of how
many files actually renamed.

for filename in *$1*   #Traverse all
matching files in directory.
```

UNIX SHELLS

Although *bash* is the default shell in Linux, there are actually many more that you can choose from

ash – A version of sh with features similar to those of the System V shell

bash – An enhanced version of *sh*, with additional functionality

csh – A shell commonly used on BSD platforms. It has a similar syntax to the C programming language

ksh – The *korn* shel.

pdksh – A public domain version of *ksh*

tcsh – Similar to *csh*, but with extra command-editing features

zsh – A version of the *korn* shell, but with additional features and compatibility with *csh/tcsh*.

For even more shells, check out http://freshmeat.net/appindex/console/shells.html

```
do
  if [ -f “$filename” ]  # If finds match...
  then
    fname=`basename $filename` # Strip off path.
    n=`echo $fname | sed -e “s/$1/$2/”`   # Substitute new for old in filename.
    mv $fname $n   # Rename.
    let “number += 1”
  fi
done

if [ “$number” -eq “$ONE” ] # For correct grammar.
then
 echo “$number file renamed.”
else
 echo “$number files renamed.”
fi

exit 0
```

SHELL RECIPE:BURN CDS

While there are many useful GUI tools for burning CDs, it's not uncommon to need to burn a CD on a server or other device where running a GUI application is a waste of resources, or simply can't be done over a serial console. The following script will pull an ISO from one data CD, store it locally, then write it to a blank CD. It is trivial to modify the script to read data from a directory, build an ISO and burn it to a CD.

```
#!/bin/bash
# copy-cd.sh: copying a data CD

CDROM=/dev/cdrom   # CD ROM device
OF=/home/david/projects/cdimage.iso   # output file
BLOCKSIZE=2048
SPEED=2   # May use higher speed if supported.

echo; echo “Insert source CD, but do *not* mount it.”
echo “Press ENTER when ready. “
read ready   # Wait for input, $ready not used.

echo; echo “Copying the source CD to $OF.”
echo “This may take a while. Please be patient.”

dd if=$CDROM of=$OF bs=$BLOCKSIZE  # Raw device copy.

echo; echo “Remove data CD.”
echo “Insert blank CDR.”
echo “Press ENTER when ready. “
read ready   # Wait for input, $ready not used.

echo “Copying $OF to CDR.”

cdrecord -v -isosize speed=$SPEED dev=0,0 $OF
# Uses Joerg Schilling's “cdrecord” package (see its docs).
# http://www.fokus.gmd.de/nthp/employees/schilling/cdrecord.html

echo; echo “Done copying $OF to CDR on device $CDROM.”

echo “Do you want to erase the image file (y/n)? “  # Probably a huge file.
read answer

case “$answer” in
[yY]) rm -f $OF
      echo “$OF erased.”
      ;;
*)    echo “$OF not erased.”;;
esac

echo

# Exercise:
# Change the above “case” statement to also accept “yes” and “Yes” as input.

exit 0
```

SHELL RECIPE:ADDING MORE SWAP

If you're in the position of having to add more swap space to a server or workstation, it can be done without the chore of repartitioning disks or having to mess around with mundane and frustrating tasks like recompiling kernels. One can create a imagefile on an existing filesystem – which can be formatted as a swapfile – then the **swapon** can be used to add it to the current swap pool.

By using this script below, we can specify how large we want the swapfile to be in blocks, with each block being 1KB. Of course, one can modify the script to write the swapfile to multiple locations, allowing multiple temporary blocks to be added to the system on-the-fly.

```
#!/bin/bash

# Creating a swapfile.
# This script must be run as root.

ROOT_UID=0         # Root has $UID 0.
E_WRONG_USER=65    # Not root?

FILE=/swap
BLOCKSIZE=1024
MINBLOCKS=40
SUCCESS=0

if [ “$UID” -ne “$ROOT_UID” ]
then
  echo; echo “You must be root to run this script.”; echo
  exit $E_WRONG_USER
fi


blocks=${1:-$MINBLOCKS}  #  Set to default of 40 blocks,
          #+ if nothing specified on command line.
# This is the equivalent of the command block below.
# ———————————————————
# if [ -n “$1” ]
# then
#   blocks=$1
# else
#   blocks=$MINBLOCKS
# fi
# ———————————————————
```

EXIT CODES

Successful execution... or not?

When a script exits, we can pass an integer back to whatever executed the script signifying if the script executed successfully and if not, the reasoning for the failure. If we exit with a **0**, then it is considered to have executed without any problems, otherwise the value that we pass back is available for error output for handling a workaround for any problem that occurs.

One key use for the exit value is to chain shell commands, or to test for a specific value in a response. When we compile code, we may do:

```
# ./configure && make && make test && make install
```

Each process along the way must return **0** for the next to execute, so if **make test** fails, it won't try to install the code we just compiled. The **&&** signifies a logical AND test, so if the first fails, the rest never get executed. We can also use a logical OR statement, or ||, so if the previous execution failed, it will execute the next.

We can expand these tests into **if** statements, where we can inspect a variable, the output from another script or binary, or the state of a file. The most basic **if** statement checks for values within a string:

```
#!/bin/sh
if [ $STR = “Test”]; then
  echo Test
else
  echo Fail
fi
```

Unlike many languages, bash does not use **==** to compare strings instead simply using the **=** operator. One can also use the **-eq** option to perform an **equals** check:

```
if [ $STR -eq “Test”]; then

if [ “$blocks” -lt $MINBLOCKS ]
then
  blocks=$MINBLOCKS   # Must be at least 40 blocks long.
fi


echo “Creating swap file of size $blocks blocks (KB).”
dd if=/dev/zero of=$FILE bs=$BLOCKSIZE count=$blocks  # Zero out file.

mkswap $FILE $blocks   # Designate it a swap file.
swapon $FILE   # Activate swap file.

echo “Swap file created and activated.”

exit $SUCCESS
```

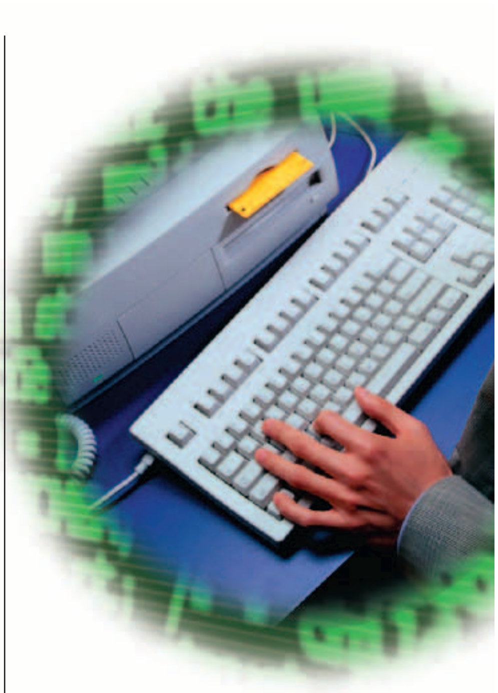
We can also add a RAM disk on-the-fly, using a similar script.

```
#!/bin/bash

E_NON_ROOT_USER=70   # Must run as root.
ROOTUSER_NAME=root

MOUNTPT=/mnt/ramdisk
SIZE=2000   # 2K blocks (change as appropriate)
BLOCKSIZE=1024   # 1K (1024 byte) block size
DEVICE=/dev/ram0   # First ram device

username=`id -nu`
if [ “$username” != “$ROOTUSER_NAME” ]
then
  echo “Must be root to run \”`basename $0`\”.”
  exit $E_NON_ROOT_USER
fi

if [ ! -d “$MOUNTPT” ] #  Test whether mount point already there,
then   #+ so no error if this script is run
  mkdir $MOUNTPT   #+ multiple times.
fi

dd if=/dev/zero of=$DEVICE count=$SIZE bs=$BLOCKSIZE  # Zero out RAM device.
        # Why is this necessary?
mke2fs $DEVICE   # Create an ext2 filesystem on it.
mount $DEVICE $MOUNTPT
# Mount it.
chmod 777 $MOUNTPT   # Enables ordinary user to access ramdisk.
        # However, must be root to unmount it.

echo “\”$MOUNTPT\” now available for use.”
echo 0
```

SHELL RECIPE: PID FINDER

Whenever there is a strange or unknown process running on a system, it's generally not a smart idea to just destroy it before we spend some time finding out what exactly it does. This script will use data within /proc to locate the binary on the system so we can investigate if it's something we need or not. This also proves useful on a compromised system, where processes may have been killed and restarted using a replacement binary in a different location, or where the system masks the real process through a trojan **ps**.

```
#!/bin/bash
# pid-identifier.sh: Gives complete path name to process associated with pid.

ARGNO=1  # Number of arguments the script expects.
E_WRONGARGS=65
E_BADPID=66
E_NOSUCHPROCESS=67
E_NOPERMISSION=68
PROCFILE=exe

if [ $# -ne $ARGNO ]
then
  echo “Usage: `basename $0` PID-number” >&2  # Error message >stderr.
  exit $E_WRONGARGS
fi

pidno=$( ps ax | grep $1 | awk ‘{ print $1 }’ | grep $1 )
# Checks for pid in “ps” listing, field #1.
# Then makes sure it is the actual process, not the process invoked by this script.
# The last “grep $1” filters out this possibility.
if [ -z “$pidno” ]  # If, after all the filtering, the result is a zero-length string,
then   # no running process corresponds to the pid given.
  echo “No such process running.”
  exit $E_NOSUCHPROCESS
fi

# Alternatively:
#   if ! ps $1 > /dev/null 2>&1
#   then   # no running process corresponds to the pid given.
#     echo “No such process running.”
#     exit $E_NOSUCHPROCESS
#   fi


if [ ! -r “/proc/$1/$PROCFILE” ]
# Check for read permission.
then
  echo “Process $1 running, but...”
  echo “Can’t get read permission on /proc/$1/$PROCFILE.”
  exit $E_NOPERMISSION
# Ordinary user can’t access some files in /proc.
fi

# The last two tests may be replaced by:
#    if ! kill -0 $1 > /dev/null 2>&1
# ‘0’ is not a signal, but
       # this will test whether it is possible to send a signal to the process.
#    then echo “PID doesn’t exist or you’re not its owner” >&2
#    exit $E_BADPID
#    fi

exe_file=$( ls -l /proc/$1 | grep “exe” | awk ‘{ print $11 }’ )
# Or       exe_file=$( ls -l /proc/$1/exe | awk ‘{print $11}’ )
#
# /proc/pid-number/exe is a symbolic link
# to the complete path name of the invoking process.

if [ -e “$exe_file” ]  # If /proc/pid-number/exe exists...
then      # the corresponding process exists.
  echo “Process #$1 invoked by $exe_file.”
else
  echo “No such process running.”
fi

exit 0

```



»

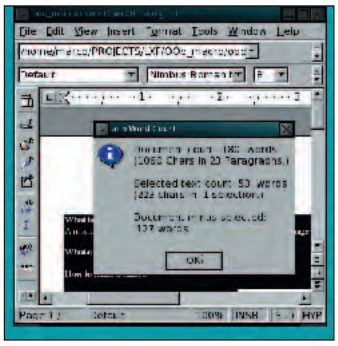
WRITING AND USING OPENOFFICE.ORG MACROS

Process office documents faster by making power tools for Linux's most popular Free office suite.

Many of its users probably don't know it yet, but the Linux office suite *OpenOffice.org* (**www.openoffice.org** – often abbreviated to *OOo*) comes by default with an internal programming language – called StarBasic – which can automate any sort of task. A StarBasic piece of code able to perform one of these tasks on demand is normally called a *macro.* Macros can really speed up your usage of *OpenOffice.org*, make it fun and sometimes get rid of (or hide) some of the suite's most annoying 'features', or make up for a lack thereof. An example? Look at **Fig1**!

No more clicking inside three levels of menus: thanks to a macro from Andrew Brown, we were able to check – almost in real-time – the word-count of this feature, both on a sectional level, and then in its entirety when *LXF*'s Production Editor put it all together.

What you probably love most about macros is that you don't need to learn yet another programming language to get advantage of them – unless you want to, of course! As **Fig2** shows, many macros are distributed embedded in *OpenOffice.org* documents which explain what they do, point to the license, and offer an 'Install' button: click on it, restart *OpenOffice.org* and your macro is available in the Tools>Macros menu (more on this later).

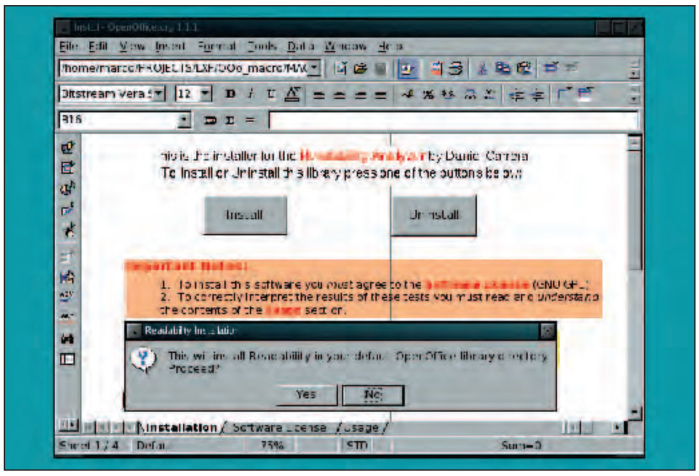

File Edit View Insert Format Tools Window Help
/home/marco/PROJECTS/LXF/OOo_macro/ooo
Default
Nimbus Roman No
Word Count
OK
Page 1 / 1
Default
100%
INSRT
HYP

Document count: 180 words
(1060 Chars in 20 Paragraphs.)

Selected text count: 53 words
(222 chars in 1 selection.)

Document minus selected:
127 words

Fig1 **Andrew Brown's word-count macro: an open-'n'-click operation.**



This is the installer for the Readability Analyzer by Daniel Carrera
To Install or Uninstall this library press one of the buttons below

Important Notes:

1. To install this software you must agree to the Software License (GNU GPL)
2. To correctly interpret the results of these tests you must read and understand the contents of the Usage section.

This will install Readability in your default OpenOffice library directory. Proceed?

Fig2 **Installing *OpenOffice.org* macros: also an open-'n'-click procedure.**

Even when a ready-made macro doesn't exist, you might be able to create it without having to resort to coding. Suppose that, for emphasis, to make particular points stand out from the surrounding copy, we would like to put just some sentences (not whole paragraphs) in Nimbus Roman font, size 8. In such cases the solution is not the Stylist, since it acts at the paragraph level, but the 'Record Macro' feature (still in Tool>Macros, *OpenOffice.org* version 1.1.x!). We highlight the sentence to format and select that option, causing a 'Stop Recording' button to appear. Next, we would hunt down all the proper menu choices in the various formatting menus (consoled by the thought that this is *"for the last time!"*), and then push the 'Stop Recording' button when we have completed our choices. *OpenOffice.org* transforms our clicks into StarBasic code (which most users will have no need to look at, and never will – why should they?) You will then be prompted to name the resulting macro, as shown in **Fig3**.

No matter how they entered your system, macros can be launched in several ways. They can be run behind the scenes by other macros, or plainly selected in the Tools>Macros window. This is very slow, however, and only recommended for macros you use very seldomly. When you select an existing macro, the same window seen in **Fig2** adds an 'assign' button, which allows you to bind the macro to (new) menu entries, combination of keys or events. In this context, 'event' means either some file-level action (like opening it) or using any piece of a form that is embedded in an *OpenOffice.org* document (see **Fig4**). This is particularly useful when the document using the macro must be distributed to many people who don't want or need to learn some other *Emacs*-like keystroke ballet.

Show me the code!

Enough theory? OK, but let's just throw in some terminology first: it not only looks more professional, but can help give you a greater understanding of the documentation on the subject. Macros can be organised in *modules* (usually sets of single-purpose procedures with similar scope) which are then assembled in *libraries*; the latter can be either available in any file you edit with *OpenOffice.org*, or inside just one of them (for instance, the 'How_To_Use_Basic_Macros.sxw' entry in **Fig3**). The usual Tools> Macros window also has an Organizer function to rearrange macros the way you want. Another thing to keep in mind when studying code or tutorial is that StarBasic is structured around subroutines and functions. The latter return values (as in "they might be assigned to variables"), while subroutines don't.

Now that the terminology is out of the way, in the spirit of the best Unix traditions, here we have our very own "Hello World" macro:

```
Sub HelloWorld
  MsgBox HelloWorldString()
End Sub
Function HelloWorldString() As
String
   HelloWorldString  = "Hello Linux
Format users" & Date
End Function
```

The subroutine **HelloWorld** calls a popup window (shown in **Fig5**). The argument of this window is the text returned by the **HelloWorldString** function: in our example, the result is built concatenating (with the **&** operator) a constant string and the result of another, built-in, function.

Let's move to a more common office situation: imagine that all the customer orders that are not processed yet are in the same folder, one per file. What if your boss asked for a daily report showing all the pending orders? Should you really do the *Dilbert* thing: that is, stop to list, cut and paste the folder content? Of course not. Just bind the following code to some key, and let *OpenOffice.org* list the files for you.

```
Sub Listfiles
   Dim sFile As String, sList As String
   Dim sResult As String
   sResult = "Files present on " &
Date & ":" & chr(10) & chr(13) &
chr(09)
   sList = "/home/marco/"
   sList = InputBox ("Enter folder to
list","Directory listing macro", sList)
```

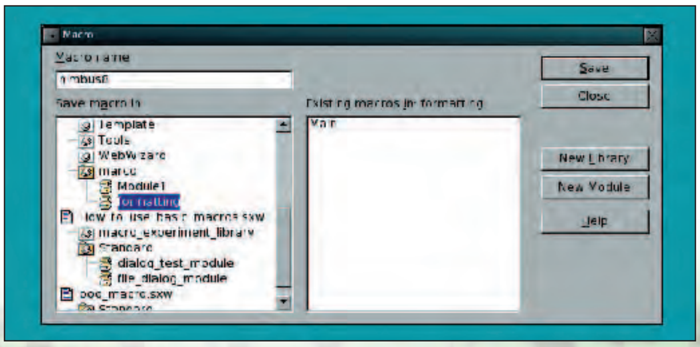


Fig3 **Create, save and organise macros without ever looking inside them.**

```
  sFile = Dir$( sList &
getPathSeparator() , 0)
  If sFile <> "" Then
  Do
    sResult = sResult & sFile &
chr(10)& chr(13) & chr(09)
    sFile = Dir$
    Loop Until sFile = ""
  End If
  MsgBox sResult
  oViewCursor =
ThisComponent.getCurrentController
().getViewCursor()
  oText = ThisComponent.getText()
  oCursor =
oText.createTextCursorByRange
( oViewCursor.getStart() )
  oText.insertString
( oCursor.getStart(), sResult, true )
End Sub
```

When we run this macro in the folder storing all the screenshots for this article, we will see this listing:

```
Files present on 05/08/2004:
  ooo_macro_helloworld.png
  ooo_macro_record.png
  ooo_macro_wordcount.png
```

After declaring and setting the necessary variables, the **ListFiles** subroutine asks (notice the **InputBox** statement) which folder it should look into: the initial value of **sList** is presented as default. In the next line, the list of all files contained in the sList directory is assigned to sFile. The following loop appends all file names, plus some formatting, to the **sResult** variable. The **chr()** functions translates ASCII values to characters: **10**, **13** and **09** respectively correspond to **Line Feed**, **Carriage Return** and **Tab**. As soon as the loop is finished, another **MsgBox** shows what we have got. The last four lines of the subroutine mean, in plain English: *"get the text part of the document I'm looking at, find where the cursor is, and place the sResult string in just that position".*

Boilerplate paradise

Have you noticed how some sentences – or groups of sentences – cannot be delivered as separate files, but need to be copied time and again inside longer documents? Things like *"Dear customer, to have your money back please follow this simple150-step procedure"* or *"My beloved $GIRLFRIEND, you are the only light in my eyes..."*. The default *OOo* solution to this problem is to write everything into a file and use the Insert>File dialog window to select it. On the other hand, why search and select every time? Write and format it once, save it into a file, and insert it with just one keystroke with a macro like this:

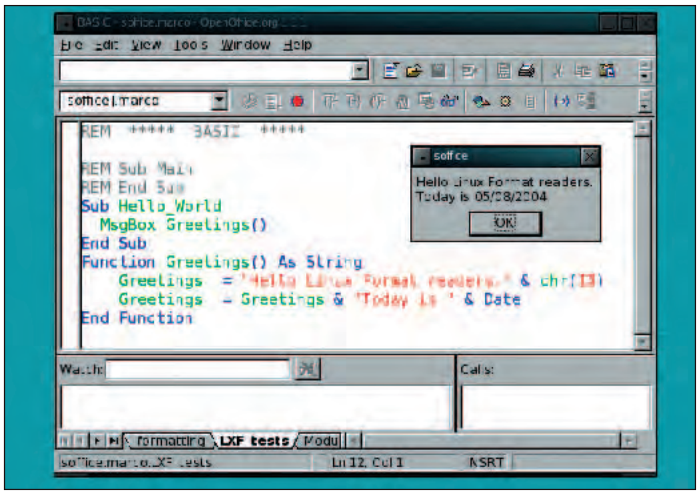
```
Sub Insert_Happy_Birthday
  'Insert selected file inside current
Writer document
  dim oDoc as object, oTextCursor
as object, oViewCursor as object
  dim sFile as String
  sFile = "/home/marco/happy_
birthday.sxw"
  oDoc = ThisComponent
  'Lock document controllers to
prevent screen updates
  oDoc.LockControllers
  oViewCursor =
oDoc.CurrentController.getView
Cursor()
  'Create text cursor so that
document can be inserted
  oTextCursor = oDoc.Text.
createTextCursorByRange(oView
Cursor.getStart())
  'Insert document at text cursor
position
  sFile = ConvertToURL (sFile)
  oTextCursor.InsertDocument
FromURL (sFile, Array())
  'Ensure text cursor is at end of
document
  oTextCursor.gotoEnd(false)
  'enable screen updates
 oDoc.UnlockControllers
End Sub
```

This code works more or less like the one in the previous example. The main difference is that the text and its formatting are read from a file with the **'InsertDocumentFromURL** method. Notice also how the document is temporarily frozen during this operation – this is done with the **LockControllers** statement.

Macros are not just for text

Macros in *OpenOffice.org* are in no way restricted to just the management of text in purely text documents. The spreadsheet in **Fig4**, is the ComboBox_in_a_Sheet_En.sxc file found in the **Ooomacros.org** website (see *Resources* box, above right). Here is the underlying code:

```
 Set oCellRangeSource =
oSheet.getCellRangeByName("B14:B
20")
 'extract the data from the range
```



```
REM  *****  BASIC  *****

REM Sub Main
REM End Sub
Sub Hello_World
  MsgBox Greetings()
End Sub
Function Greetings() As String
    Greetings  = "Hello Linux Format readers." & chr(13)
    Greetings  = Greetings & "Today is " & Date
End Function
```

Hello Linux Format readers.
Today is 05/08/2004

Fig5 **Everybody's first OpenOffice.org macro - with traditional message.**

MACRO RESOURCES

The three best macro resources for *OpenOffice.org* end users are without doubt the tutorial by Andrew Pitonyak available at **www.pitonyak.org/oo.php** and the two portals **www.ooomacros.org and www.ooextras.org**. The word-count macro is maintained by Andrew Brown (**www.darwinwars.com/lunatic/bugs/oo_macros.html**). With the exception of the "Hello World" example, the other macros mentioned in this article can be found in these places, or were obtained by that code with some minor editing. The *OpenOffice.org* site offers a *How to use basic macros* tutorial **http://documentation.openoffice.org/HOW_TO/various_topics/How_to_use_basic_macros.sxw**. Should you want to get serious about macros, there's a good Programmer's Tutorial at **http://api.openoffice.org/basic/man/tutorial/tutorial.pdf**.

»

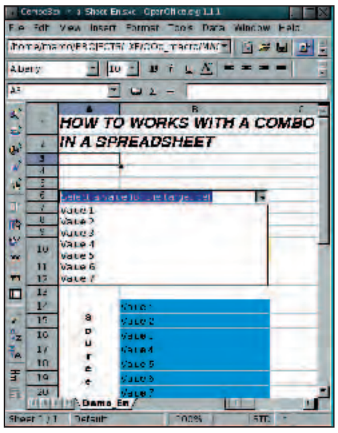


Fig4 **Running macros through an HTML-like form.**

«

```
    aSourceDataArray() =
oCellRangeSource.getDataArray()
  ....
      For I = LBound(aSource()) To
UBound(aSource())
      aDataArrayRow()=
aSourceDataArray(I)
      aSource(I) = aDataArrayRow(0)
  Next I
    oComboBox.StringItemList =
aSource()
```

In the first line, we get the cells containing the values to choose from (the blue ones in the picture). The rest of the code makes an array of strings with their content (**aSource**), which is eventually placed inside the combo box. Please note that this and the other snippets of code in this article, have been cut to fit in these pages. For the complete source code, see the *Resources* box on the previous page.

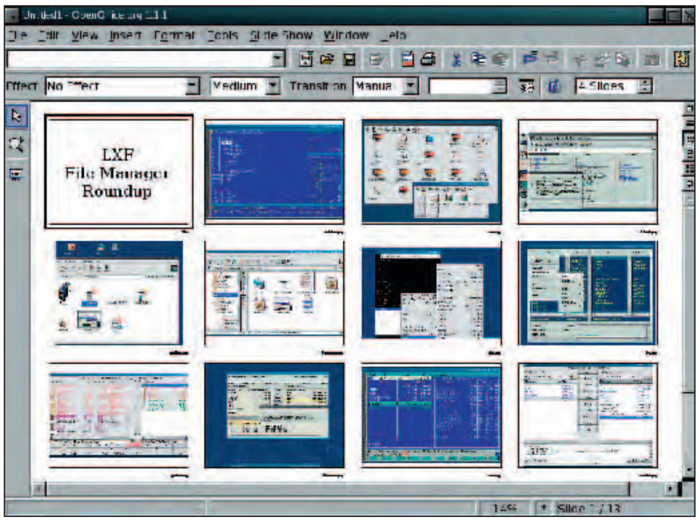


Fig6 **Macros can make your image management a more automatic process.**

We haven't forgot about images and presentations – you can make a full slideshow of your vacation with a thumbnail macro. Enter the name of the folder with the pictures, and then just enter the title in the first page.

PROGRAMMING WITH DCOP

If the idea of making KDE programs talk to each other seems too advanced, you're exactly the sort of user who will be excited by this section...

For many people, programming is some kind of black art practised by those stereotypical sandals-and-beard types that use their cryptic languages for creating programs and websites. Although this banality could certainly contain a modicum of truth in some parts of the programming world, there are some other uses for coding: places where you don't need to know your compiler from your pre-processor and your **byte code** from your **#ifdef**. One such area is the Desktop Communications Protocol (DCOP).

DCOP is a special system that was created by the KDE development team to provide a means for apps to be able to communicate with each other. The need for this technology was twofold. First, hackers of KDE software needed a way to communicate with each other (such as loading web addresses into *Konqueror* automatically) and secondly to provide users with a means of scripting their desktop to work exactly as they want it to. *LXF* is currently featuring a KDE Development series (turn to page 74 for the latest instalment), but to prove there is life outside *KDevelop*, here we are going to focus our efforts on using this DCOP technology for tweaking and scripting out KDE desktops.

Getting to grips with DCOP

Metaphorically, any KDE application such as *Kate*, *Konqueror, KOrganiser (*or any other KDE application) can be likened to a black box. This black box runs on your system, and unless you are a programmer who can understand the source code, the black box basically works as a single unit and you cannot do anything particularly exotic with it. With DCOP, the developers that created this black box added some cables and plugs that allow you to plug one black box into another. DCOP acts as a method of taking information from one program and pushing it in somewhere else. With this capability, we can do some clever things with our desktop.

These DCOP 'cables' and 'plugs' are made available in the form of functions. Each function does something specific that is useful. As an example, the **getClipboardHistoryItem** in the *klipper* tool will get a particular item from the clipboard history. Although the concept of these functions is fairly simple, how do you know which functions are available?

KDCOP is a tool that displays everything that DCOP can muster. This lesser-known utility is started by typing **kdcop** into the Run Application box (**Alt-F2** in a default KDE) or into a terminal. When the program loads, you are presented with a list of applications in a tree. These are all applications that are currently open on your system. When you expand each application, you will see additional categories of DCOP functions. These different categories contain functions for different parts of the application.

You can test functions in KDCOP by clicking on them. As an example, load the *Konqueror* browser and you will see it added to the tree. You will see the entry as something such as konqueror-3211. The numbers in the tree entry refer to the process ID of application. This is appended so we can use DCOP functions on specific processes. If you expand the *Konqueror* tree and then click on the **KonquerorIface** entry you should see a function called **DCOPRef createNewWindow(QString url).** If you click on this function and type in a web address, a new window will pop up displaying the website.

Scripting DCOP

One of the benefits of DCOP is that you can make use of these functions to write your own programs that can tie different KDE programs together to do different things. This script-ability is available in four main languages; C++, Python, Perl and shell scripting. We are going to cover shell scripting as our preferred language, as we can get it up and running quickly. All of our programs will use the commonly used *bash* shell. You can check which shell you are using by typing:

```
echo $SHELL
```

Never done any shell scripting before? Don't worry – we are going to use some fairly simple features of the *bash* shell, some of which were mentioned earlier on in this feature, and we will explain everything in the scripts and how it works. In the next few pages you will not only be learning how to use DCOP and *bash* scripts, but you will also be learning how to bring your different KDE applications together to do something useful. As is true with any form of programming, practice makes perfect, and we recommend that you explore the web for different examples of useful little DCOP and *bash* scripts.

OUR FIRST DCOP PROGRAM: A DICTIONARY SHORTCUT

We are going create a useful little script that will allow us to highlight some text and check its meaning on **dictionary.com**. This script will need to do a few things to work:

- Find out what has been selected
- Load up a *Konqueror* window
- Pass the selected word to **dictionary.com**

This is actually a fairly simple program to write. To ease us gently into our first example of DCOP programming, we will explore each step of the process and explain it fully. This should help in cementing our initial understanding of how to connect those different DCOP cables and plugs together.

Creating a script

These DCOP shell scripts are created as plain text files. To kick off, we will create a file called dictionary.sh with the following code in it:

```
#!/bin/bash

echo “This works!”
```

To make this code actually run as a script, you will need to enable the execute bit in the file. This can be done with the **chmod** command:

```
chmod a+x dictionary.sh
```

You can then run the script from within the same directory with:

```
./dictionary.sh
```

Write the code

Shell scripts are typically littered with normal commands that access other programs on the system; this is the same with DCOP. If we wanted to run a DCOP command from the shell (such as *Konsole* or an *xterm*), we can use the **dcop** command to specify the program and its DCOP function. As an example, we can do this with the very first part of our program. We need to use the *klipper* program (the little clipboard in your KDE system tray) to tell us what is the most recent selection made. Whenever you select something with your mouse, it will be added to the *klipper* menu, so we can use a *klipper* DCOP function to find out what this is. This function is called **getClipboardHistoryItem** and you can run it from your command-line terminal thus:

```
dcop klipper klipper
getClipboardHistoryItem 0
```

Here we use **dcop** to specify the *klipper* program, then the *klipper* sub-category within the *klipper* program, the **getClipboardHistoryItem** function and we finally pass **0** to this function. The **0** means that we want the first (latest) selection in the menu (remember that in programming speak, many languages begin counting at **0**). If we wanted the third item, we would specify **2**. When you run this command, you will see the latest selection returned displayed.

We now need to put this command in the script and put the result (the latest selection) into a variable. Add this to your script below the **#!/bin/bash** part (you can remove the **echo** command we added to the script earlier):

```
NAME=`dcop klipper klipper
getClipboardHistoryItem 0`
```

Here we are storing the result of the command in a variable called **NAME**. Notice how we need to specify the command between backticks (` `); if we didn't use backticks, the command would not be executed as a whole. To make this even easier, you can use KDCOP to write the code for you. Simply find this function by expanding the *klipper* application, expanding the *klipper* group and then drag the **getClipboardHistoryItem** function into your text editor where you are writing the script. You will see the complete command added for you, and you simply need to set the **i** at the end of the line to **0**.

We now have the latest selection in the **NAME** variable and we simply need to pass this selection to **dictionary.com**. The question is, how on earth do we tell dictionary.com what we want to search for without typing it into the search box? Again, the solution is fairly simple. If you go to **dictionary.com** and type in a word such as “guitar,” you will get your expected dictionary definition appearing. Have a look at the address bar and see what it says: **http://dictionary.reference.com/search?q=guitar**

As you can see, the word is simply added at the end of the web address. You can test this by editing the search word directly in the address bar, such as this example: **http://dictionary.reference.com/search?q=drum**

»

All we need to do now is to launch *Konqueror* with the **dictionary.com** web address and the contents of our **NAME** variable delicately balanced at the end of the address. Although you can launch *Konqueror* with DCOP, it is actually easier to just launch it the normal way; type in **konqueror** and the web address that you want the browser to load:

```
konqueror http://dictionary.reference.
com/search?q=$NAME
```

Here you can see that we have referenced our value by putting a dollar sign (**$**) before the variable name. When this script runs the **$NAME** part will be replaced wit the contents of the variable (the latest selection). Our complete script now looks like this:

```
#!/bin/bash
NAME=`dcop klipper klipper
getClipboardHistoryItem 0`
konqueror http://dictionary.reference.
com/search?q=$NAME
```

If you run your script now, you will see everything working together and displaying your latest selection at **dictionary.com**. Try making some different selections and re-running the script. To fully complete the process you should now create an icon on your desktop that runs the script, or ideally create a keyboard shortcut. This would give you the opportunity to select a word and press a simple key combination to instantly get a dictionary definition. This could be useful for those occasions when you stumble across an overly complicated word on a web page. When creating keyboard shortcuts, you should always make sure that you reference the script with a full path (such as /home/jono/dcopscripts/dictionary.sh) or place the script in your path.

As an additional exercise, you could now modify the script to make a second one that points to **thesaurus.reference.com** instead of **dictionary.reference.com** to provide yet another useful script that looks up a selection in an online thesaurus.

SELECTION WORD-COUNT

Our next script is going to push our flexibility with DCOP that little bit further. We are going to throw some new ingredients into our mixing bowl, and these open up an endless amount of potential in your scripts. These ingredients are some new commands and *kdialog*.

Adding other commands to our scripts opens up the potential to bring in special functionality that may not be a part of KDE. A typical example is the command that is used to count the number of words in a file (**wc**). The **wc** command is used like this:

```
wc file.txt
```

When you run this command on file.txt, you will be presented with the following output as an example:

```
6   34  234
```

This output provides three types of information. The first number is the number of lines in the file, the second is the number of words and the third is the number of characters. This information is quite useful for those people who need to write a limited amount of words. The only problem with **wc** is the fact that it is really intended to work on files.

One feature that we at *LXF* are always keen to see in word processing applications is the ability to run a word-count on a selected number of words. This functionality is present in most modern proprietary word-processing applications, but for some reason, several FOSS word-processing apps in the past haven't included it by default. This function was essential while writing this coding feature, as it has different sections that needed to be a certain length. It would be great if we could highlight the text and run **wc** on it. Currently, we need to copy the text and put it into a separate file and then count the words using **wc**. This is a problem that DCOP can solve.

Finding a solution

To solve this problem, we first need to select the text to be counted as we did in our previous script. This is simply a case of using our *klipper* line from our previous script:

```
SELEC=`dcop klipper klipper
getClipboardHistoryItem 0`
```

The next step is to run the **wc** command on our selection and separate out the three different numbers into different variables. Before we actually separate out the numbers, we need to discuss how to run **wc** on the selection as opposed to a file. This is performed by using the *bash* **echo** command to output the selection and pipe it into the **wc** command. You can test this by running this command in your shell:

```
echo The cat sat on the mat | wc
```

You will see that this command correctly outputs the information we need. We can now use

```
echo $SELEC | wc
```

to run this on our selection. To incorporate this and separate out our three numbers into variables, we need to run the following command:

```
set -- $(echo $SELEC | wc);
LINES=$1; WORDS=$2; CHARS=$3
```

In this line, we use the *bash* **set** command to set the **LINES**, **WORDS** and **CHARS** variables to the correct values. To do this, we run the command in brackets first and then the three different separated numbers are set to the appropriate variable. **LINES** is the first number (**$1**), **WORDS** is the second (**$2**) and **CHARS** is the third (**$3**).

Displaying the results

With the bulk of the work out of the way, we now only need to display the contents to the user. Although we could theoretically just output this to the shell, the idea is that we bind this script to a key binding or icon again, so we really need a GUI output. This is where *kdialog* steps up.

The *kdialog* program simply gives us a dialog box that we can display information in. There are lots of different types of dialog box (such as those with yes/no buttons, error boxes, information boxes, drop down selection boxes *etc*). You can find a list of the available types of dialog box by running:

```
kdialog --help
```

We are going to use the **-msgbox** type of dialog (this gives us an information box). To use it, we simply specify what we want to appear in the box in quotes. We can also use **\n** to split the output over new lines:

```
kdialog --msgbox "Selection
details\nWords: $WORDS\nLetters:
$CHARS"
```

That's it. Our clever little selection word-count dialog is complete. Here is the completed code:

```
#!/bin/bash
SELEC=`dcop klipper klipper
getClipboardHistoryItem 0`
set -- $(echo $SELEC | wc);
LINES=$1; WORDS=$2; CHARS=$3
```

TIPS AND TRICKS

1 Unsure about a particular command-line option? Most programs have some help available when you pass **--help** as a switch.

2 You can include more than one command on the same line, separating them by a **;**.

3 User's home directories are refereed by ~user-name/, and the current users home directory is under ~/

4 Looking for that misplaced file? locate uses the daily filesystem database build to search for files containing the pattern you pass to it. **locate foo** would list all files with foo in their path.

5 ls isn't particularly helpful with default settings, so add **--color=tty** to it and you get pretty colours, depicting the type of file. Simply add an **alias** to /etc/profile, **alias ls='ls --color=auto'** and it'll do it all the time.

```
kdialog --msgbox "Selection
details\nWords: $WORDS\nLetters:
$CHARS"
```

As you can see, when you bring in an external command (such as **wc**) and use *kdialog* to output information, the potential for DCOP increases dramatically. Any command-line command that you can think of can be theoretically used to work with DCOP. As an example, there's no reason why it wouldn't be fairly straightforward to use the festival speech synthesis system (**www.cstr.ed.ac.uk/projects/festival/**) to speak your selections. This would be a handy hack for Linux systems that are likely to be used by people suffering from visual impairments.

THE BIG ONE: BLOGLINES NOTIFIER

Now you've got a bit of confidence, we are going to explore how to use some clever *bash* scripting to check if a blog reading list is updated. The reading list in question is a website called **bloglines.com**. You can register on the site for free and add people's blog RSS feeds to your account. The clever thing about **bloglines.com** is that you can download a little notifier that lets you know when there are new blog entries to be read. We are going to write such a notifier here.

Checking for updates

To check for updates on bloglines, we need to visit the following address:

http://rpc.bloglines.com/update?user=username@address.com&ver=1

In this instance, we are using the **username@address.com** address to specify the account. When you visit this URL you will get something such as the following in your browser:

```
|0||
```

Here, **0** indicates the number of messages to be read. This value can also be **-1** if there is a problem with the username (such as it not existing on the system). As you can see, the number of messages is nestled inside some pipes (|); these pipes are used to separate the information reported. The first task is for us to cut out the number of messages and put it in a variable:

```
RESULT=$(echo $(lynx
"http://rpc.bloglines.com/update?use
r=username@address.com&ver=1"
 -dump | awk -F'|' '{print $2}'))
```

Here we use the *Lynx* text web browser (see this month's *Roundup*) to visit the site and use the **-dump** switch to output the page outside the browser. We then run it through the *awk* engine that cuts out the number of messages. We haven't got the space to discuss *awk* in detail here, but we use the field separator switch (**-F**) to set our separator as the pipe (|). We then print out the second entry (**$2**) as our output. All of this results in **RESULT** being set to the number of messages.

Displaying the results

With our notifier, there's three possible outcomes based on the contents of **$RESULT** that we can display:

-1 – This means a bad configuration. The username may be incorrect or not exist.

0 – There are no messages, so there is no point displaying the bloglines website.

Anything else – any other number is the number of unread messages. We should specify the number of messages and display the **bloglines.com** website if needed.

We can choose the right course of action by using the *bash* shell case structure. In this structure we open a case block and specify which value we are testing:

```
case $RESULT in
```

Here we are testing our **$RESULT** variable. We can now use a series of stanzas to specify the action, based on the contents on **$RESULT**. Our first is if there is a bad configuration:

```
 -1)
 kdialog -msgbox "Bloglines
Notifier\nError in configuration.
Maybe a a bad username."
 ;;
```

The first line specifies the value we are testing (**-1**) behind a right bracket. Anything on the following lines will be run if **$RESULT** contains **-1** until it reaches two semi-colons (**;;**). Here, we only have a single line to be run: our familiar **kdialog** command.

The following two case statements should be fairly straightforward (see the code at the end of this page). The final case value (*****)) applies to everything else that may be in **$RESULT**; this applies to unread messages. Next, we check a variable at the beginning of the script to see if we should open a *Konqueror* window when there are messages so the user can read the messages. In this final case block, we use the *bash* **if** statement to check if the **$OPENPAGE** is set to **1** (this means to open the browser). If this is set to **1**, we open *konqueror* and set the website as the main Bloglines website.

Here, in its full and final glory, is our completed script:

```
#!/bin/bash

# For yes - 1, For no - 0
OPENPAGE=1
RESULT=$(echo $(lynx
"http://rpc.bloglines.com/update?use
r=username@address.com&ver=1"
-dump | awk -F'|' '{print $2}'))

case $RESULT in
 -1)
 kdialog -msgbox "Bloglines
Notifier\nError in configuration.
Maybe a a bad username."
 ;;

 0)
 kdialog -msgbox "Bloglines
Notifier\nNo messages."
 ;;

 *)
 kdialog -msgbox "Bloglines
Notifier\n $RESULT message(s)"
 if [ $OPENPAGE = 1 ]; then
   konqueror
http://www.bloglines.com/myblogs
 fi
 ;;
esac
```

> **"Don't wait for someone to write a full program when coding on the command-line can produce an elegant solution..."**

In this example, we are pulling together some useful command-line tools and some useful KDE tools to create a customised script for our needs.

You should now have a sense of the true potential of DCOP, KDE and the Linux command-line tool set. There are *unlimited numbers of possibilities* – we suggest you look further into the tools discussed here. You could find an elegant solution sitting on your hard disk without having to wait for someone else to write it as a full program... LXF

END OF STORY?

The examples and information collected here have hopefully opened your eyes to the opportunities that exist to write your own script and dabble with code. We have plenty of programming tutorials in every issue of *Linux Format*, and we're always open to suggestions on new tutorials or features on any Linux-related subject – mail linuxformat@futurenet.co.uk with your comments or ideas.

What on Earth is the... O(1)Scheduler?

Ever feel you do too much at once? Jon Kent spares a thought for your CPU...

Before you start, what exactly is a scheduler?

Within Linux there are two types of scheduler, the I/O scheduler and the process (or task) scheduler. With the 2.6 kernel, there are two types of I/O scheduler to choose from: the deadline scheduler, and the anticipatory scheduler. However, with process scheduling, there is only one scheduler, and that is the O(1) scheduler, which we will be examining here. From this point, we will be using the term "scheduler" to refer to the O(1) scheduler unless otherwise mentioned.

In a nutshell, the O(1) scheduler selects which processes to run at a given time on the available processors. As Linux is a multitasking operating system, the scheduler has to split up the available resources – *ie* processor time – between all the processes that are waiting to run. You need to bear in mind that although it may appear that all of your processes are running concurrently, in actual fact *only one process can run on one processor at any given time*. Therefore, one of the major tasks of any scheduler is to ensure that it is utilising the available resources efficiently thereby given the impression that all processes are running concurrently.

So how does the scheduler manage this?

Like most modern operating systems, Linux is now a pre-emptive multitasking operating system. Until the release of the 2.6 kernel, the 2.4 or older kernel.org kernels were not pre-emptive, although there was a patch available written by Robert Love that provided this for 2.4. Pre-emption means that a process is allowed to run until such time as the scheduler deems it necessary to suspend the process – or in other words pre-empt the process. The amount of time that a process is given is called a timeslice, and is dynamically calculated by the scheduler. The dynamic nature of this timeslice allows the scheduler to manage the resources available to it based upon the current utilisation. This allows it to easily react when the system is under heavy load.

This seems simple enough, but how does Linux decide which process to run?

Like other UNIX systems, Linux provides the concept of scheduling classes. Within Linux there are two scheduling classes, these being real-time (RT) and timeshare (TS). Some other UNIX systems implement additional classes, for example Sun Solaris implements four scheduling classes; these being real-time, timeshare, interactive (IA) and system (SYS). Within Linux, most processes will run within the timeshare class. Within these classes, Linux implements process priority, and in addition provides dynamic priority-based scheduling. We need to cover both of these concepts to gain a good understanding. Process priority is relatively straightforward. Within Linux there are two priority ranges used: the 'nice' value set by the process or user, and the dynamic priority. Hopefully you have an idea of what a nice value is, but to recap, the nice value ranges from -20 to 19, with a default of 0. The lower nice value a process has the higher priority it has. Therefore a process with a nice value of -15 has a higher priority than one with a nice value of 2.

To complicate things a little, the scheduler can impose a value of -5 to +5 to the initial nice value, which is called the dynamic priority. Any additional value is imposed based upon the interactivity of the process. The more interactive the process, the lower the value of the dynamic priority. This allows the

scheduler to share resources effectively. The schedule uses this nice value – and any imposed value – to decide which process to run. Should two processes become runnable at same time, the scheduler will run the process with the lowest overall value. Also, if a process becomes runnable while there is a process executing, the priority is checked, and if the process that has become runnable has a priority that is lower the current executing process, the executing process will be pre-empted to allow it to run.

How does the scheduler know if a process is interactive?

The method that the scheduler used to determine how interactive a process is, is actually surprisingly straightforward. The scheduler monitors the process to see how much time it spends sleeping, against how much time it spends running. A process that is interactive will spend a large proportion of its time sleeping, as it is waiting for something to occur, such as a keyboard entry; and as such, these processes are commonly referred to as I/O bound processes. However, if a process spends most of its time running, it is considered not to be interactive; and if it spends a nearly all of its run time runnable, it is referred to as processor-bound. To add more detail to scheduler dynamic priority method, if a process spends a great deal of its time I/O bound and exceeds its timeslice, it will not be given a high dynamic priority; whereas if the opposite is true, a lower value will be used (thereby increasing the priority).

You talked a bit about timeslicing – does the priority affect the timeslice value?

Very much so. As mentioned, the timeslice is dynamic not static, and is determined by – initially – the priority of the process to be run, and then the dynamic priority from that point on. The dynamic priority can change between each execution of the process, so the timeslice allocation will change to reflect this. This change in timeslice allocation is recalculated once all tasks in the current active queue have been completed and moved to the expired queue, which we'll examine more later.

The lower the value – remembering the lower values have higher priorities – the longer the timeslice that is set. However, should the timeslice expire, the process will still be pre-empted by the scheduler. Conversely, should the process complete executing before the timeslice expires, it does not have to wait until it does expire to release the resources back to the scheduler. In addition, to ensure fairness and to avoid underhand practices, if a processes forks a child both parent and child processes share the remaining timeslice. After the timeslice has expired, the parent and child will then be assigned separate priorities and run separately.

The minimum timeslice that the scheduler will assign is 10 milliseconds and the maximum is 200 milliseconds. For a process that has just be created, it is assigned initially a timeslice of 100 milliseconds as the scheduler cannot known the interactivity yet of the new process.

Ah, but there always seem to be exceptions...

How cynical of you, but you are correct – there is an exception. As we briefly outlined, Linux also provides real-time scheduling – in addition to timeshare – should a process require it; and there are two possible real-time policies that can be used. These are SCHED_RR (round robin) and SCHED_FIFO (first in, first out). The difference between these is that SCHED_FIFO real-time processes can run until they have completed: in other words, there is no timeslice. Whereas SCHED_RR processes are restricted by a timeslice. In addition, with real-time processes, there is no dynamic priority value assigned by the scheduler. The scheduler will always ensure that when a real-time process becomes runnable, it is executed.

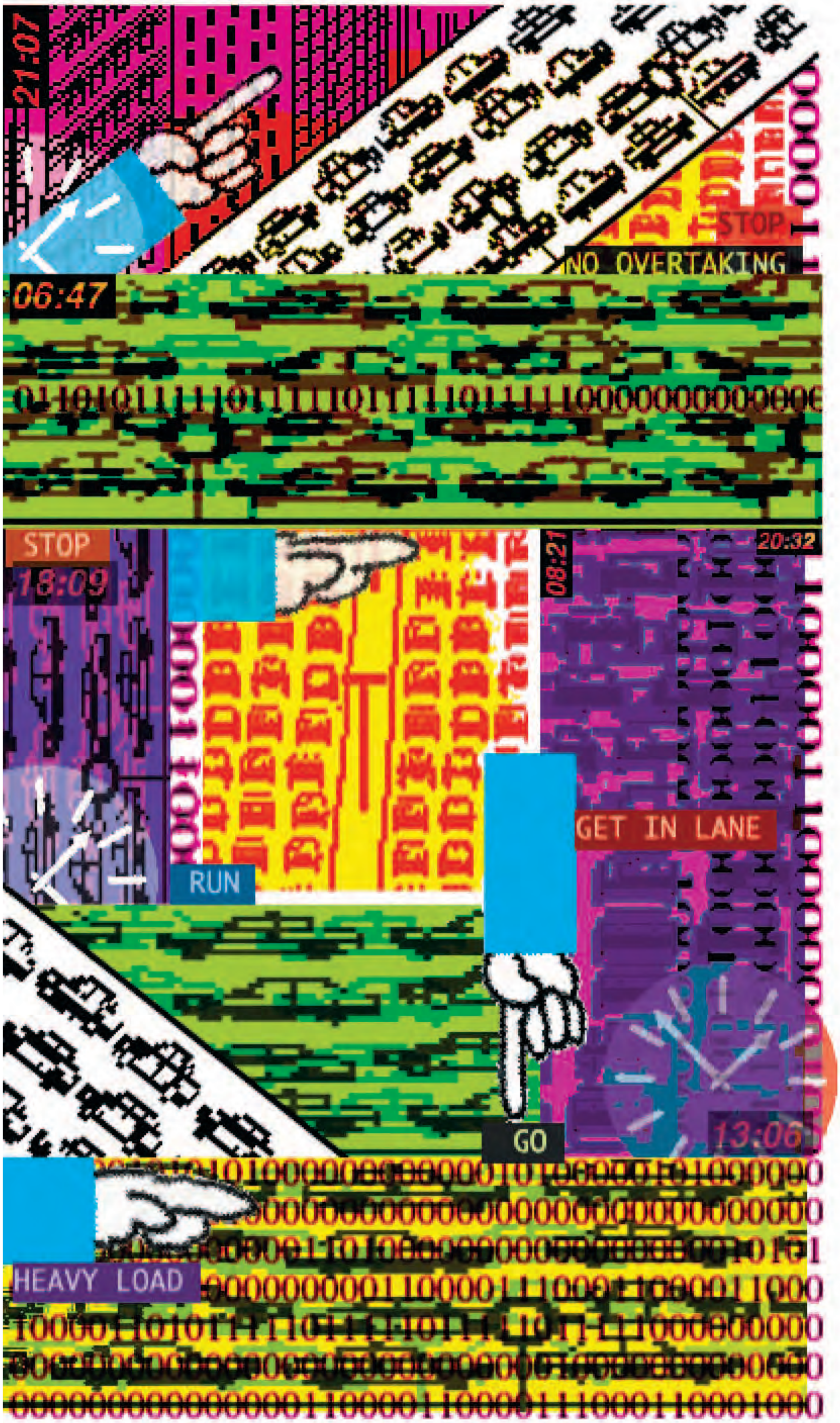
Although there is a real-time scheduling class within the O(1) scheduler, this is not truly real-time, but provides 'soft' real-time. This means the scheduler will try to schedule a process within the timelines expected, but may not always be able to achieve this. There are patches available to provide 'hard' real-time that guarantee to meet scheduling requirements, if this is required.

Great! So, I can run *Mozilla FireFox* in real-time, then?

Well you could, but you would find that your system slows down by a large degree, as you are not allowing the scheduler to manage your system effectively. Real-time priority is only required by latency, or time-sensitive applications such as some video capture applications or market data delivery systems. After all, you would notice if you missed a large number of frames when video capturing; and if you where trading off market values you would care if you lost data or received delayed data, as this may lead you to miss an opportunity in the market. However I'm sure you could survive if *FireFox* took a few milliseconds longer to render a page! However, there are some ways that can be used to increase performance, especially if you have an SMP system. All of these options require careful thought and benchmarking to see whether you are likely to achieve any real benefit.

The easiest one to implement is increasing the priority of a process within the current class. Priority escalation can be worth looking into, and is usual easiest to implement via **suid**ing the binary you wish to escalate to root permissions. The usual **suid** cautions apply here and you will want to ensure that the process in question has been written correctly. A good example of a well-written process is one that uses the increased scope of root to escalate its priority to less than 0 and then drops back down to the actual user id.

Although not 2.6-specific, a surprisingly useful tweak is to use IRQ affinity to bind your network interface cards' IRQ to a »



single CPU, which then avoids the overhead of context switching. Again sensible benchmarking needs to undertaken to ensure there are actually any benefits to using this.

Lastly there is the option of binding a process to a single CPU, referred to as processor affinity. Processor affinity, only available on SMP systems, should be viewed with caution and should only used on large SMP systems, either implemented via several separate processors or via hyperthreaded processors although results vary with HT usage. Another issue with processor affinity is that the scheduler can still assign tasks to the processor that you have assigned a process to, as there is native implementation of processor shielding – sometimes referred to as processor sets. Lastly, at present there is no *proc*-based interface to processor affinity and is only available via the syscall interface.

I think I've got that, but what's so wrong with the 2.4 scheduler?

The scheduler in 2.4 was a big improvement over 2.2 and took a straightforward approach. All processes were held on a single run queue, no matter how many processors where installed in the system. Each process was assigned a 'goodness' rating which took into account the nice value, whether the process was previously pre-empted or completed before the timeslice expired and if the process has requested CPU affinity. So when a processor became available the schedule would examine the run queue and run the process with the highest goodness value. Simple enough.

However, there were several problems. The run-queue itself was not order in anyway so the scheduler had to look at entire run-queue to find the next process to execute. The more processors you had, the longer this run-queue became, leading to longer searches of the run-queue, which impacted performance. Lastly, because of these long look-up times it is possible that two processors could choose to run the same process, meaning yet another long run-queue look-up. It was possible for the scheduler to take more time performing these functions than actually running processes. Not good.

Therefore, one of the main motivations behind the new O(1) scheduler was to improve upon the systems performance, especially SMP systems. The scheduler is designed to ensure that the time to select the next process to run, and allocate that to the processor, is constant. Unlike the 2.4 scheduler, which was affected by system load and large SMP systems, the O(1) scheduler has been designed to be unaffected by system load.

Whoa, the 2.4 scheduler sounds like it's seriously floored, I'm upgrading to 2.6 now!

Not so fast! To be fair to the 2.4 scheduler, unless you are running a box with four or more processors and a large run-queue, it is unlikely that you would have seen these issues. So, jumping to 2.6 should not be rushed into because of the known problems with 2.4 running under a heavy load. If you already have a stable system, moving to a new kernel may only provide you with limited benefits.

Does 2.6 still use a single run-queue, as the single run-queue seems to be part of the problem with the 2.4 scheduler?

The 2.6 kernel now has a run-queue for each processor and each runnable process exists on exactly one of these run-queues. The scheduling details we talked about previously are also held in the run-queue as two priority arrays: an active and expired array. It is these arrays that provides the details for the O(1) scheduling used in the 2.6 kernel. As these arrays are at the heart of O(1) scheduling, it is worth going into more detail on them.

The two main arrays, active and expired, are used to manage the processes on the run-queue. When a process has completed or exceeded its timeslice, and therefore was pre-empted by the scheduler, it is moved onto the expired queue. Once the active queue is empty, the expired queue becomes the active queue and the processes are allocated time on the processor again.

The priority arrays themselves contain a single queue, or list, of runnable processors per priority level, of which there are 140 possible priorities. This is implemented as a bitmap field that has one bit for every priority available on the system, which is initially zero. When a process becomes runnable, the corresponding bit is set to one in the bitmap. This makes finding the highest priority process on the system simply a case of finding the first set bit in the bitmap. This makes find the next process to run easy to achieve, and once the end of queue has been reached, the top of the queue is queried in a circular manner as mentioned previously

If a process is considered by the scheduler to be highly interactive, the scheduler can decide to move the process – once it completes its current timeslice – back onto the active queue, as opposed to the expired queue. The aim here is to ensure that a highly interactive process does not have to wait until the current active queue is completed and switched to become the expired queue, to become runnable again.

Ah, but I think that I can see a problem with this method. Surely it is possible for all processes to exit on one CPU leaving it idle, while the others are busy?

That would be a very real issue if the scheduling was left as outlined previously. However, in order to avoid this situation, the scheduler checks to see if CPU balancing is required, to ensure that workload is evenly balanced. This load-balancing is carried out every 200ms unless the processor is idle; in which case, this check is carried out every 1ms. If this was not implemented, then the situation you outlined would most definitely occur.

So why is the new Linux scheduler called O(1)?

The term O(1) is used to describe the new scheduler's algorithmic behaviour. The O(x) notation – also known as *big-oh* notation – is a mathematical term that refers to an algorithm's behaviour, or the algorithm's asymptotic behaviour. Asymptotic behaviour is intended to show how well an algorithm scales with greater data inputs. For the extremely curious and mathematically minded, here is a mathematical definition:

Definition (big-oh): Let f and g be functions from the set of integers (or the set of real numbers) to the set of real numbers. Then f(x) is said to be O(g(x)), which is read as f(x) is big-oh of g(x), if and only if there are constants C and n0 such that | f(x) | <= C | g(x) | whenever x > n0.

In English, this basically means that the time to complete g(x) is always greater than the time to complete f(x). To make this easier to picture, imagine you have ten processes to run, and each is run for one second on one processor. It would therefore take 10 seconds to run all processes, or in big-oh notation O(10), or O(n) for n processes.

Now back to O(1). Again, to simplify this explanation a little, imagine you have ten processes to run on one processor, but this time the processor can run all ten at the same time, and no matter how many processes you want to run the time remains the same. This is O(1) as it is constant.

Hmm, right. Its looks to me like I need to move to the 2.6 kernel to gain the scheduling benefits you have been outlining?

The answer depends upon which distribution you are running. If you run a so-called Enterprise Linux distribution, you may not need to. Red Hat Advanced Server 3.0 and associated products use the 2.4.21 kernel, but have backported a surprisingly large number of features from the 2.6 kernel series to this kernel, including the new O(1) scheduler. Whether this kernel should really be called 2.4.21 is up for debate, but if you are running this distribution you have O(1). Linus himself has said that he has no problem with backporting, so if it OK with Linus, there is no real need for a big debate on this.

If you prefer, you can also patch your existing 2.4 kernel, as there are patches available which backport the O(1) scheduler – and other 2.6 features – to the 2.4 kernel series. As with all things Linux, you have multiple choices that enable you to pick the ones that best fit your needs. However, the easiest, and probably safest route is to test out the 2.6 kernel. LXF

IMPLEMENTING...
IRQ Affinity

IRQ affinity is easy to implement, but however, it is harder to benchmark. It is possible though – here is how you can achieve it.

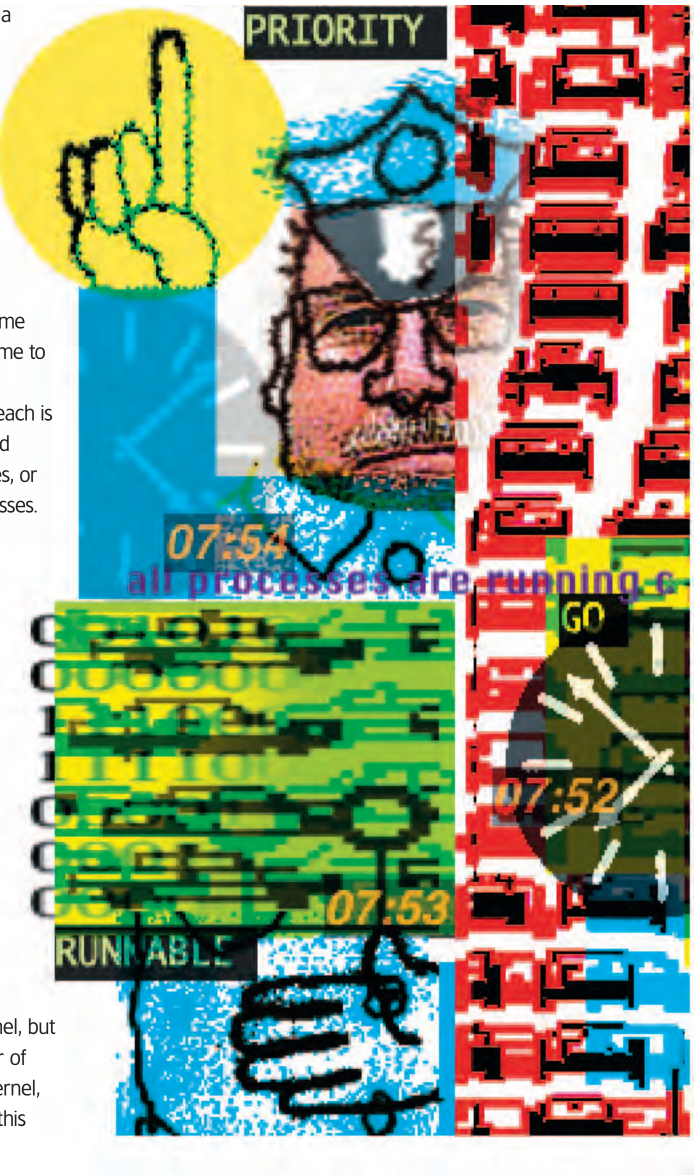
Identify the IRQ of the card you wish to use by looking at /proc/interrupts.

Once you have this, run the following:

```
echo "1" > /proc/irq/[irq number]/smp_affinity
```

In this example, 1 refers to CPU0, 2 would refer to CPU1 and so on.

To ensure this has run correctly check /proc/interrupts and ensure that the counter for the irq only increases for the CPU you have assigned that irq to. Obviously, to make this permanent, you will need to add or modify a startup file.



Tutorials

Our experts offer help and opinions on a whole host of Linux applications

YOUR GUIDE TO GETTING THINGS DONE!

Whether you are just starting out in Linux, or an experienced veteran, there's always more to learn. Every issue of *Linux Format* is packed full of practical advice, and nowhere is it more concentrated than in our tutorials. pages.

Here you'll find expert guides to all sorts of things, from Basic Linux usage to understanding and deploying network solutions, from simple script coding to the complexities of Perl regular expressions, Java server apps and more. We aim to bring a good mix of tutorials to each issue, but if you have any suggestions for topics you'd like us to cover, why not contact us, by email at **linuxformat@futurenet.co.uk** or by snail mail, or log on to **www.linuxformat.co.uk** and post your suggestions in our special forums? Hope to hear from you soon!

Nick Veitch EDITOR

HOW CODE IS REPRESENTED

Including code in magazines can be tricky, but we hope our notation will help it become clear. When lines are too long for our columns, the remaining text appears on the next line in a solid blue box:

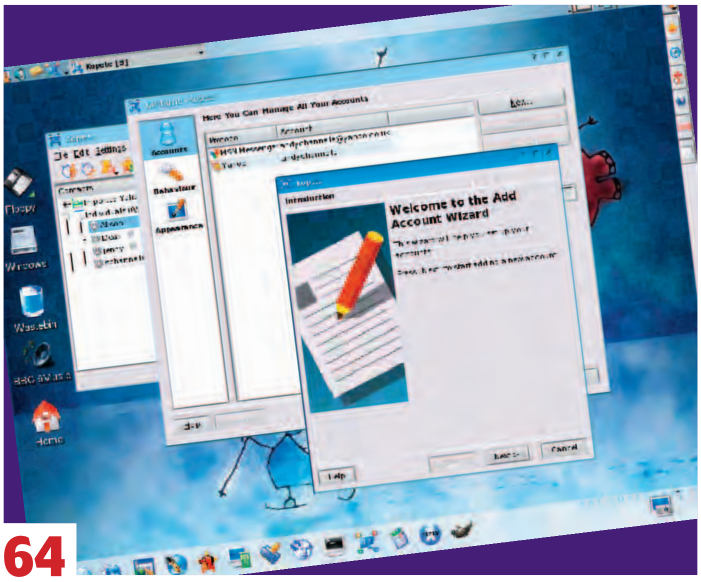
```
procedure
TfrmTextEditor.mniWordWrapClick
(Sender: TObject);
```

otherwise, there is usually a gap between lines:

```
begin
  mniWordWrap.Checked := false;
end;
```

Usually, you'll find the code on our CD/DVD too.

THIS MONTH TEACH YOURSELF...

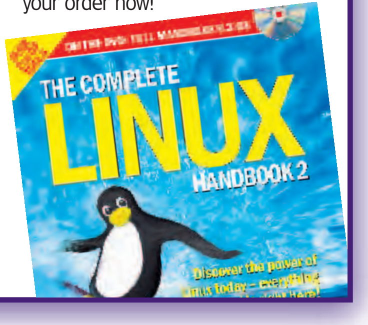


64

TIP OF THE MONTH! LINUX IN ONE

As *Linux Format* is now 55 issues young, we've seen dozens and dozens of excellent cover stories, tutorials, *What on Earths*, and other features fly through the magazine and onto the newsstands. About a year ago, we launched *The Complete Linux Handbook*, which was a compilation of many of our best tutorials up until that time, with a fairly strong focus on programming. Unsurprisingly, it sold out in less than a month, and is now a collector's item – if you're lucky enough to own a copy, keep it somewhere safe along with the included disc!

This month we've launched a second handbook, creatively named *The Complete Linux Handbook 2*, that complements the original handbook with even more *Linux Format* content. This time you'll find 260 pages crammed with some our best material, priced at £14.99. Home users will find beginners' tutorials, help working with *OpenOffice.org*, *Wine*, and *GnuCash*; as well as a wide selection of tutorials on *Blender* and *The GIMP*. There are also two large sections on networking and programming as well as a huge *Answers* section to get all your problems solved. Most importantly, it all comes strongly bound to keep it safe on your bookshelf! On the attached DVD is a full copy of Mandrake, as well as a broad selection of the most popular software from our coverdiscs to date – we expect this to sell even faster than our previous book, so get down to your local newsagents and place your order now!



BEGINNERS' GUIDE TO LINUX APPLICATIONS

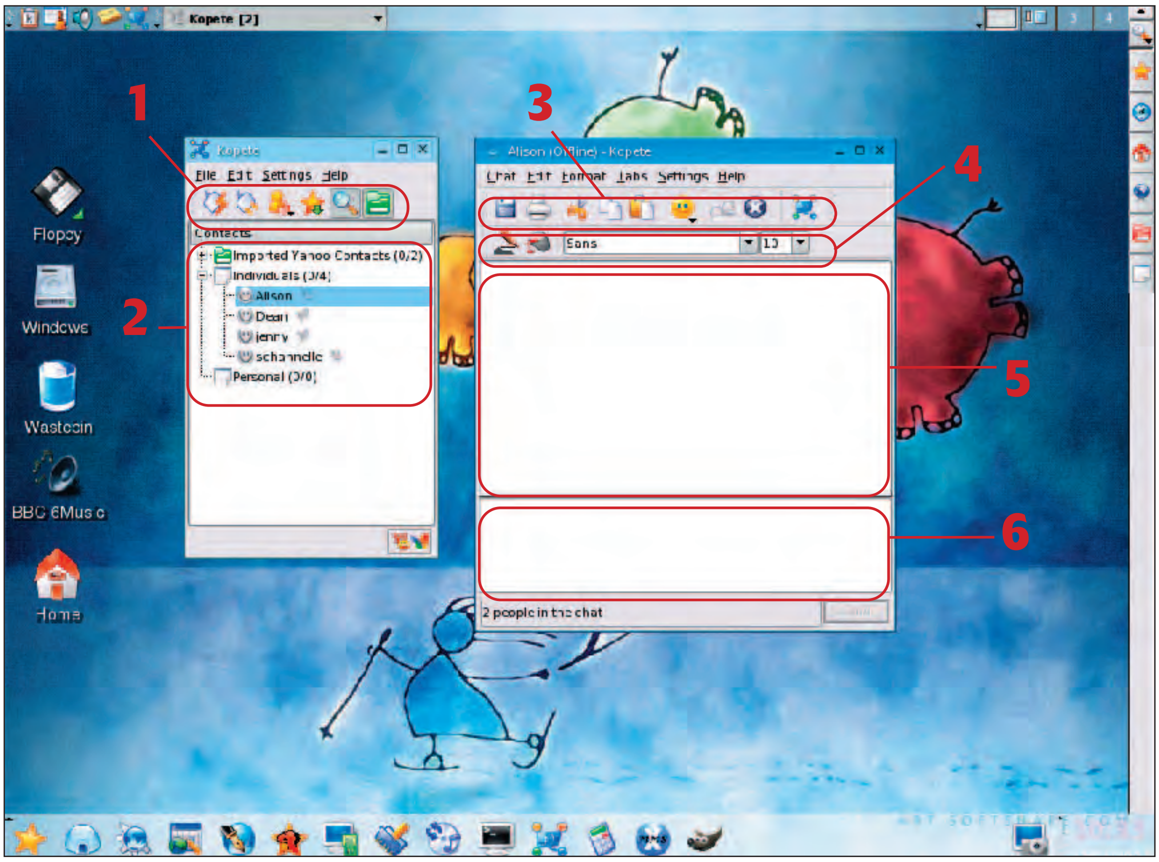
Instant Messaging with Kopete

Linux is blessed with more than one great Instant Messaging app. And if your friends can't agree on whether to use MSN, AOL, Yahoo! or something else, you needn't feel left out – Andy Channelle shows us how to chat to them all!

On the CD and DVD

Over the course of this *Beginners'* series, we have already looked at email and web browsing software – the two most popular uses of the Internet – but have yet to address the other areas of Net communication: Instant Messaging and FTP. This issue, we'll be looking at the various Linux applications available for the former, with a promise to cover FTP servers and clients next time.

You may have noticed a pattern in the world of Linux development. That is, one team of hackers works on an application for KDE, while another team works on a GNOME equivalent. While this may seem like trying to invent the wheel twice, it does provide for a degree of friendly competition driving development forward; and as most of these projects are Open Source, when one team comes up with a killer feature, it can quickly be integrated into rivals. Everyone wins, especially the user! This is a long-winded way of saying there is a core GNOME Instant Messaging (IM) application, which is called *Gaim*, and a KDE one named *Kopete*; and that both have a similar feature-set. We're covering the latter, but much of the information here is pertinent to *Gaim*, and both are included on the coverdiscs for you to try.



Kopete – setup and use

Kopete is the Instant Messaging tool that comes bundled with KDE. The latest version at the time of writing is 0.8.0 and is included with KDE 3.2, but it is always worth checking out the project's website at **http://kopete.kde.org** for updates – if there is one constant in the world of Instant Messaging, it's that protocols are updated, occasionally breaking the connections of third-party applications.

There are two elements to *Kopete*, we'll call them the Connection Window (on the left in the main image) and Chat Windows (on the right). Note the use of the plural for the latter. While only one Connection Window should be open per session, there can be *any number* of Chat Windows on the desktop at any one time, as each one corresponds to what is happening in an ongoing conversation. Fans of uncluttered desktops (well, those using KDE3.2!) can pick up a small but very useful applet called *Universal Kopete*, which transfers this Window into the Universal Sidebar from **www.kde-apps.org.**

The Connection Window is what appears when we first launch the application, and it is from here we can organise network connections and contacts and configure global settings such as the way the Chat Windows display their content.

1 Connection Window Toolbar

This bar contains six tools, which are all quite self-explanatory:

Connect All Clicking on this icon will cause *Kopete* to attempt to connect to all configured networks. Unless the application has been set to retain passwords (and the password aspect of *Kopete* seems a little willful) it is necessary to key in the relevant security stuff. To disconnect or connect to a service individually, use the Network Connection icons (see below) instead.

Disconnect All Simply shuts down all IM network connections. Click to remove yourself from the Instant Messaging world!

Status Status can become terribly important in intense IM chats, because not responding for a period of time can seem like a snub. With this we can take a break by selecting 'Set away globally' and choosing either Busy, Gone or Custom Message. The latter, of course, gives you the option of stating exactly why we are not available. Clicking on this will cause all the Network Connection icons to display their busy status and the appropriate message will be sent across the networks. Click the button again and select 'Set available globally' or see below for advice on setting availability on individual networks.

Add Contact Obviously, the whole point of Instant Messaging is that we have people to message instantly. So, we add the details of our contacts with this button and then, when they come online, their status is betrayed in the Contacts list (see below). Clicking on the button will start the Add Contact Wizard. If our friend's details were in KDE's Address Book we could select the radio button at the base of the dialog and choose an entry from there, but as they're not in the address book, we must do everything manually. By the way, if you are using existing accounts which already have contacts set up (*ie* Yahoo!) these will be automatically imported on the first connection.

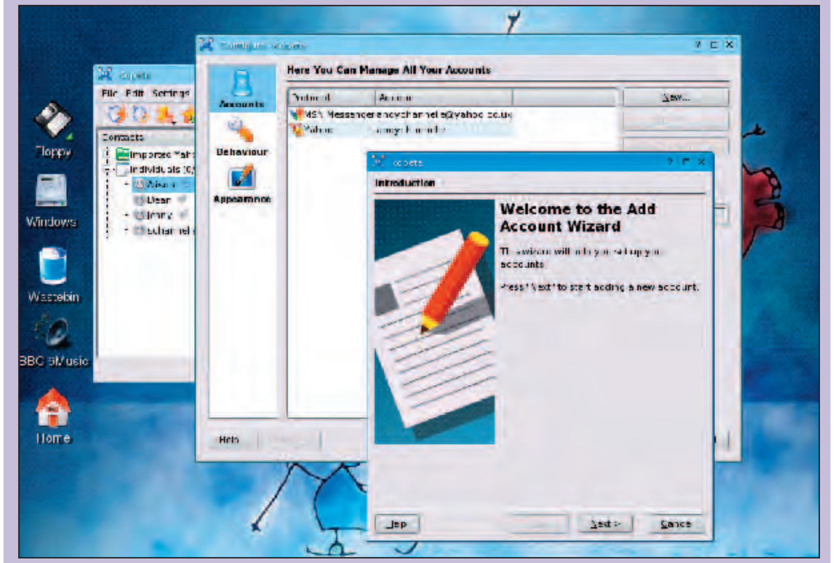
The wizard is quite simple. We enter the screen name (this is the name displayed on our computer) for our contact and then choose where the name will go in the Contacts list – personal, work, football, *etc*. We can then select the account we want to associate with this person and add their network details, that is the Yahoo! name, MSN Passport or ICQ number that they use.

Display Offline Users With this button selected all contacts will be visible regardless of their online/offline status. Otherwise, only online friend will have icons. When a contact is not online, their icon will be greyed out.

Show Empty Groups Selecting this will show up any groups you have configured that don't have contacts within them. You can create new groups by doing 'File>Create New Group...' and adding the details. Moving contacts around within groups is simply a case of dragging and dropping names, and groups can be removed or renamed using the right-click context-sensitive menu. »

CREATING ACCOUNTS...

Signing up to an IM service and sending SMS



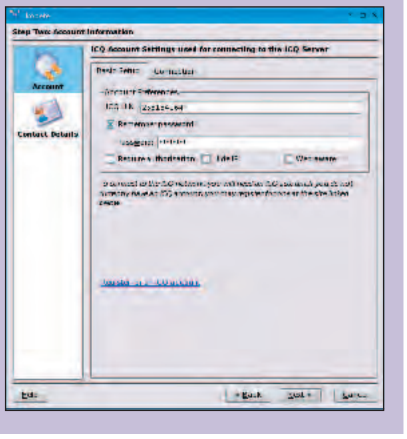
In order to use an Instant Messaging application, obviously, we first need to create accounts, and this means signing up to a service such as Yahoo!, MSN or ICQ. This can be done in one of two ways: either at the website of the respective network, or through *Kopete* itself. To do the latter do 'Settings>Configure *Kopete*...' and, in the accounts tab select 'New' to start the Account Setup Wizard.

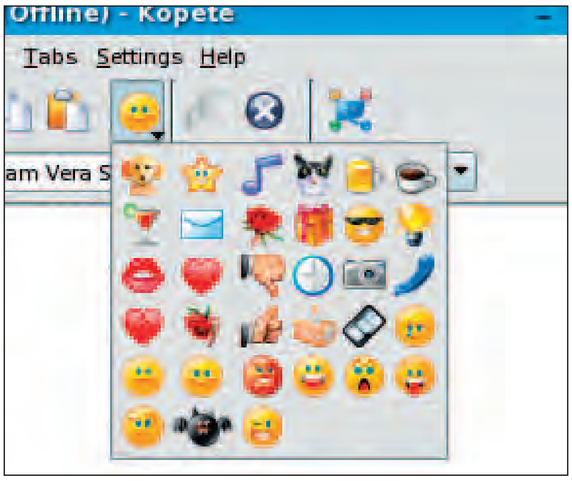
We hit 'Next' to start the process, select 'ICQ' as the 'Protocol to connect to...' and then hit next again. As we have yet to sign up for an ICQ account, we can select the 'Register...' link within this dialog to do so. This opens *Konqueror* (or other system-registered browser) at the ICQ registration site (http://go.icq.com/register/).

Once we have gone through the registration process, we can input the required information into the Account dialog, as well as any extra information – age, gender, interests, *etc* – that we may want other users to see.

Finally, when the account set-up is completed, we can see the new protocol in the bottom right of the Connection Window. An unconnected account will be greyed out, while an active one will display a colour icon.

As well as ICQ, *Kopete* also includes plug ins for AOL, MSN, Yahoo!, Jabber, IRC and SMS, so we can chat to people across different networks. As an additional bonus for users of the most recent version, it is possible to send SMS (Short Message Service) messages to any mobile phone. Just right click on the ICQ icon and select 'Send SMS'. This will bring up a small dialog with space for a number and message; complete both and hit 'Send SMS'.


Step Two: Account Information
ICQ Account Settings used for connecting to the ICQ Server
Account
Contact Details
Remember password
< Back
Next >
Cancel



Icons can sometimes convey your intentions better than text alone.

« It is possible – and even sensible, if you're planning to spend a lot of time using it – to alter this toolbar to suit your own way of working. Simply go into 'Settings>Configure Toolbars...' and make the adjustments. The dialog is a familiar looking two-paned one with available buttons on the left, current set up on the right, and a set of arrow keys in the centre. To add a button, highlight it on the left and click on the **right** button. You can then alter its position on the toolbar with the **up** and **down** arrows.

2 Contacts list

This is our (small) circle of online friends organised into groups. At its simplest we can double click on a name in this list to send a message. If that person is online, the Chat Window will open up ready for action. If they're offline and the protocol supports it, anything we send will appear on their screen next time they log in.

All the other options regarding group/contact management are accessed via the right-click, context-sensitive menu. The basic context menu contains entries for starting a chat, sending a message, changing the details of the entry and altering its position within the contacts window. The last entry on this menu contains network-specific options so, for instance, if we were chatting with Jenny via MSN we could choose to send a file directly to her, view her MSN profile, or 'block' her from future chats.

Network connections

Ranged along the bottom of the window is a set of icons representing networks to which we are currently connected. In our case we have Yahoo!, MSN, ICQ and Jabber configured and ready to use, but MSN and ICQ are both 'offline', as denoted by their monochrome icons. We can connect to either by right-clicking on the icon and selecting 'Set online' or 'Online' respectively.

Beyond connecting and disconnecting, these icons can be used to indicate our status to other users, using either the messages provided by the service or – in the case of both Yahoo! and ICQ – by setting a custom message that will be sent out to other users. There are other services occasionally hidden in here, such as Jabber's 'Join group chat' which is the IM equivalent of a chat room, and ICQ's 'Send SMS' applet.

3 Chat Toolbar

Now we're on to the space where actual conversations take place. As with the first toolbar, we can do 'Settings>Configure Toolbars...' to add or remove icons here to reflect the way we like to work. As standard we have the usual file and clipboard icons before getting to the first *Kopete*-specific icon: the emoticon or 'smilies' button. Clicking on this will open a flyout, with a selection of icons to augment your typed conversations. Select an icon to insert it into the text.

Icons don't match your desktop theme? This can often be the case in Linux applications that have been developed in isolation from the evolving look-and-feel of the desktops, but *Kopete* has this base covered. Go back to the Connection Window and do 'Settings>Configure Kopete...' and select the 'Appearance' option. You can now choose from a variety of different emoticon styles, with the 'Default' and 'KMess' sets being the most complete. However, remember that choices here only affect the look within your window, not that of the person on the other end of the conversation – and that using an unsupported emoticon will just

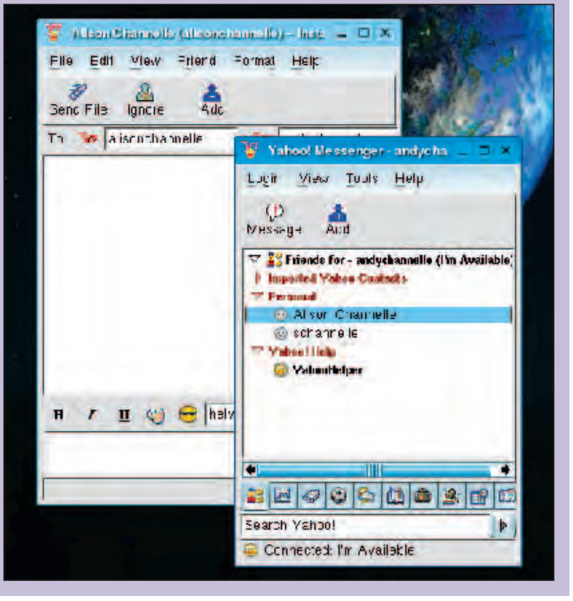
YAHOO! MESSENGER

Going native with Linux Instant Messaging

Yahoo's Messenger application has been available for Linux for years. It is not updated as often as the Windows client, but on the occasions when *Kopete* (and others) have difficulty connecting to Yahoo's servers, it is a useful backup. Of course this is only relevant if your friends are on Yahoo and not some other network.

Should you be interested in using it, you can pick up the application from http://messenger.yahoo.com/messenger/download/unix.html and versions are available for Red Hat (RPM), Debian (.deb) and FreeBSD (.tgz). RPM's are available for RH 6, 7, 8 and 9 and we had no problems installing the latter version on machines running Mandrake (9.x and 10.0), Fedora and SUSE with their respective application management tools.

Once installed, you can open the 'run' dialog and type /usr/bin/ymessenger to launch the application. Under SUSE, the installer added an entry to the *KMenu* under the 'Internet>More programs>Yahoo! Messenger'. In use, it's very similar to *Kopete*, although in testing we found that the emoticons didn't always survive when conversing between Linux and Windows XP versions of the software.



Yahoo! is the only one of the big three IM networks to supply a native Linux client.

send its textual equivalent to your chat-buddy (the Heart will send (L), the lips send a (K), *etc*). More emoticon sets are available from **www.kde-look.org**.

Next on this toolbar is the Tabs icon. By default this will be unavailable as *Kopete* is set up to open individual chat sessions in a new window. To change this, go back to the main settings ('Settings>Configure Kopete...') and select the 'Behaviour' option. The relevant section is 'Chat Window Grouping Policy.' There are various options here and choosing any of the 'Group...' entries will ensure the tab icon is active upon hitting the 'Apply' button. We can now have lots of concurrent chats going on without cluttering up the desktop. Clicking on the icon will detach the currently selected chat tab from the main window and give it one of its own.

4 Format

This toolbar changes the way our text looks, with the first button affecting the colour of the text and the second its background colour. Both will launch KDE's default colour selector. The third and forth elements are the text font and size selection dropdowns. Like the emoticons, this is really for the benefit of the local user, though some of the protocols do support the sending of coloured text.

5 Conversation pane

This is where the conversation is built up as text is entered by the various members of the chat group. We can adjust the way text is displayed here – go to 'Settings>Configure Kopete...' in the Connection Window, select Appearance and have a play - and also save conversations for archive/journalistic/legal purposes. To do the latter, do 'Chat>Save' and give the file a name. You can relive past conversations by opening the directory in *Konqueror*, right-clicking on the file and selecting 'Open with...>KHTML.' The file will open in *Konqeror*'s main window in plain text form.

PROTOCOL PROMISCUITY

It's good to talk...

Imagine what life would have been like if, in the beginnings of the telecoms revolution, rival telephone companies refused to interoperate and you could only talk – or do business with – users with the same equipment. It's not inconceivable that the entire technological direction of the 20th Century would have been slowed without the time/space compression enabled by a universal telephone service. And yet, in the IM space, corporate interests demand that AOL users shouldn't talk to MSN users, who shouldn't talk to Yahoo! users. This insane situation see the main players adapting their protocols quite often in order to break network-independent clients such as *Kopete* and *Trillian* in the Windows world.

On one hand we might argue that it's not fair to demand businesses such as Microsoft or AOL/Time Warner to allow free access to their bandwidth for non-revenue-raising users, but on the other a more open system would benefit their customers, who would continue to provide 'eyeballs' for advertising regardless of the network preferences of their friends. Rumours persist that the three main players are locked in talks to come up with a standard (and AOL's purchase of ICQ makes this a more likely prospect) but as of yet, nothing concrete has come from it.

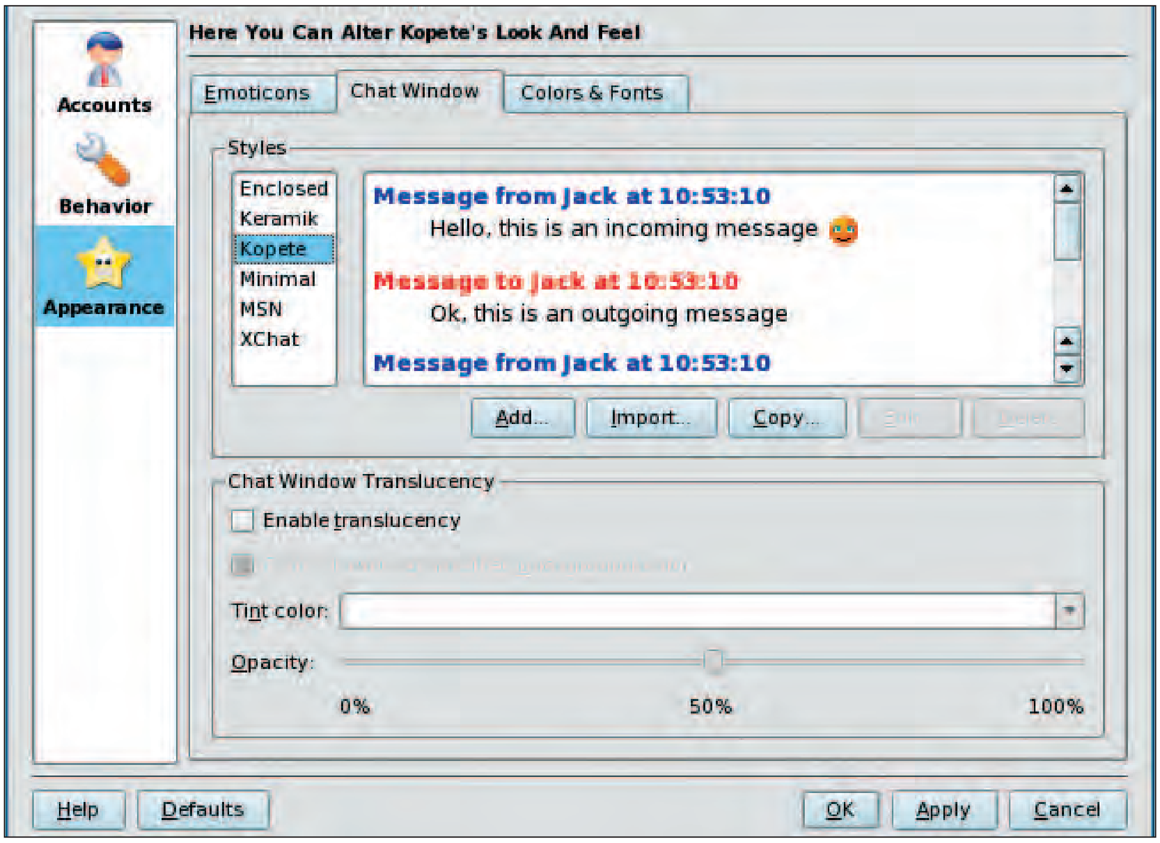
For the users of *Kopete*, *Gaim*, *et al*, this means the small inconvenience of ensuring your client is up to date, as fixes are usually released within hours or a few days of a protocol adjustment – just keep an eye on your client project's homepage.

For instance, we have had persistent problems connecting to Yahoo's network with *Gaim*. This may be the result of a recent change to the company's IM protocol and it could be fixed by the time you read this. Check the project's website (http://gaim.sourceforge.net/) for the most recent update.

We will endeavour to cover *Gaim* in some detail when we tackle the joys of video conferencing with *GNOME Meeting* in a future *Beginners'* tutorial.

6 Text input pane

Click an insertion point in here, type some stuff and hit the 'Send' button... That's it! To complicate matters, we can enter emoticons using the button on the toolbar, but the basics of Instant Messaging are very simple. We type something, send it and it appears on our correspondent's screen. They then type something and it appears on ours... LXF



Here You Can Alter Kopete's Look And Feel

Styles: Enclosed, Keramik, Kopete (selected), Minimal, MSN, XChat

Message from Jack at 10:53:10
Hello, this is an incoming message

Message to Jack at 10:53:10
Ok, this is an outgoing message

Message from Jack at 10:53:10

Chat Window Translucency

□ Enable translucency

Tint color:

Opacity: 0% 50% 100%

You can change the way conversations are displayed, but for some IM services, your recipient will not see the changes.

NEXT MONTH

Next time, we're staying on the network, but we'll be file sharing instead of chatting with the help of *Kbear, gFTP, BitTorrent* and others. Please let us know via the *LXF* forum (www.linuxformat.co.uk) or directly by email (andy@channelle.co.uk) if there's anything that you think we should be covering in this series.

GETTING TO GRIPS WITH DIRECTORIES

Server School: LDAP

Implementing directory services? Nick Veitch looks up LDAP.

On the CD and DVD

What is LDAP? It stands for *Lightweight Directory Access Protocol*. The important and salient word here is 'directory'. A basic explanation would encompass the fact that it is a service that stores data in a structured way, is indexed and can be searched or queried by client applications.

The details of LDAP are described in RFC2251 if you care to read up on it at **www.faqs.org/rfcs/rfc2251.html**, but here we're considerably more interested in getting it to work and seeing what it can do for us.

You may be asking yourself whether a directory isn't just a database by another name. To a large extent that assumption is partially true, in that they both store and retrieve structured data; but there is a difference in approach, mainly due to the type of data that needs to be stored. Most of the information stored in a directory is pretty static – like usernames, telephone numbers, addresses, passwords, for instance. They do change on occasion, but in the order of weeks and months, rather than seconds and minutes.

The nature of the data stored in both, and the way it is stored is different too. Database entries really bear little relation between each other, except for relationships that you may wish to impose. On the other hand, directory information is extremely structured and hierarchical. This is the real key to a directory – it offers a simple, fast and easily managed way of storing deeply structured

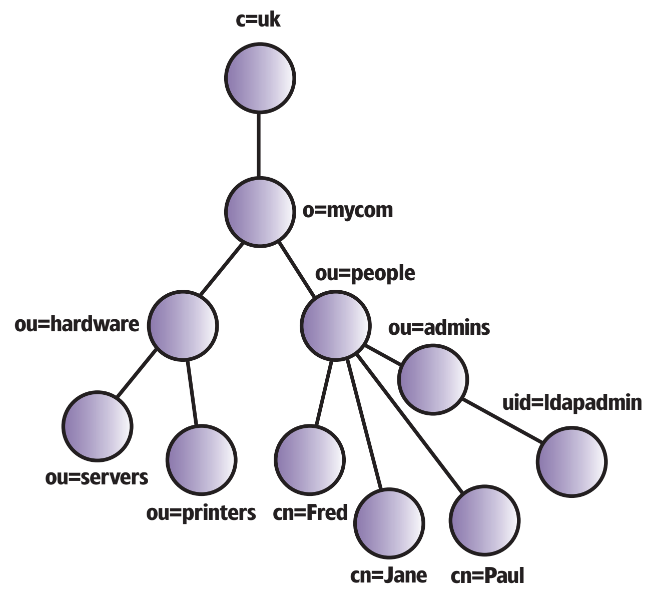


FIG1 TRADITIONAL HIERARCHY

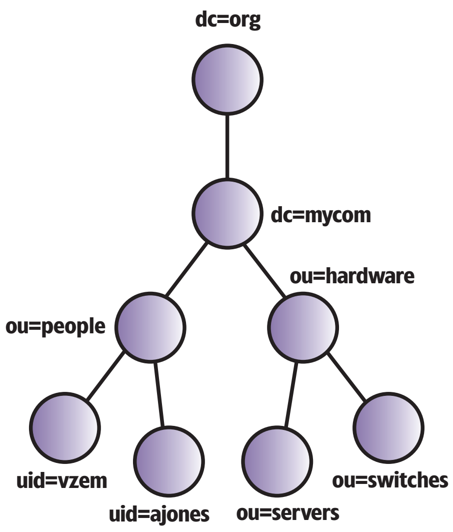


FIG2 INTERNET NAMING HIERARCHY

information. The root DN (or Distinguished Name) of a directory establishes a scope, and additional objects and containers are placed on a tree that very much resembles a filesystem. This information structure is flexible too. The labels in a directory entry can – by necessity – often have more than one value: something unthinkable in a database.

For example, an entry for a person may have more than one telephone number (increasingly important these days, most of us have more than one telephone – home, work, etc). In a database, this information would have to be stored in a separate field, which would also have to be present in all the other records too – a waste of resources.

Although it is most often used as a directory of company employees, there are practically limitless instances in which LDAP can be used – it could contain a hierarchy of hardware assets for example, or a list of parts.

Organisation

The key to the directory being quick is the organisation of the data. The directory is structured as a tree of data. Every entity contained within it can then be referenced by tracing its 'branches' back to the 'root'. The root itself is the only element which is essential for the directory to run – this gives the directory its sense of scope, and particularly when dealing with many directories, prevents all sorts of confusion. The initial Distinguished Name (or DN) is the root of the tree, and all objects it contains will reference it.

The DN of an object, for example:

```
cn=Nick Veitch,ou=people,dc=lxf,dc=org
```

establishes its place in that tree. This has some interesting effects – it means, for example, that there can be two entries for **Nick Veitch**, as long as they don't conflict in the hierarchy.

In the days when directories were first invented, there was some idea that directories would use company names and countries of origin for there roots. So we might have **o=Linux Format**, **c=uk** as the DN of an LDAP directory (see **Fig1** on the left for an example of this structure). Unfortunately, this scheme got into bother when it became clear that many directories may end up with the same root DN, especially if they had the same main name. Having two directories with the same root DN is not a good idea (especially when we come to look at referrals). So a new system was devised using something which would be unique – DNS entries. Using a corporate domain name as the basis of the root DN guarantees uniqueness (see **Fig2**). However, you are likely to come across a mix of schemes.

There are many directory implementations, and indeed, a quick Google search will turn up many versions of LDAP too. For our purposes though, we'll stick to one of the most popular ones, *OpenLDAP* – on the *LXF* coverdiscs. It doesn't have all the extra features boasted by some directory software, but it is well maintained and conforms to the standards.

Installing Open LDAP

Before you think about using *OpenLDAP*, you will most likely need a few other bits of software, depending on how you want to use it. An *SSL* transport layer is required for LDAPv3 compliance, and is worthwhile having – install *OpenSSL* (**www.openssl.org**) if you can. You'll also need the *Cyrus SASL* (*Simple Authentication and Security Layer*) **http://freshmeat.net/projects/cyrussasl/**, and there may be other authentication services you wish to use (for example, *Kerberos* – **http://web.mit.edu/kerberos/www/**).

A further consideration is database software. Although we said that a directory is a specialised database, *OpenLDAP* actually uses a standard database for its backend. The LDAP protocol is effectively a layer between the client applications and the actual data. *OpenLDAP* can use a variety of backends. The default is to use the *Sleepycat Berkeley DB*, which is freely available and probably came with your Linux distribution. It is also possible to use backends such as *DB2* and *MySQL*, if that's what you require.

The next, obvious, step is to retrieve the latest *OpenLDAP* source. This tutorial is being written using the 2.1.30 stable build that is on the *LXF* coverdiscs, but it is frequently updated, so your best bet is to get a fresh copy from the *OpenLDAP* site or one of its mirrors (see **www.openldap.org** for a list of download sites).

After unpacking the archive, you will need to configure the software. *OpenLDAP* uses *autoconf* to create a **makefile**, so this should all be very familiar by now. The default options will build a working *ldap/slapd* combination, but if you want to enable other options, such as a different database or to enable replication (*slurpd*), you will have to run the configure script with the appropriate options. Check out which options are available in the readme or by using:

```
./configure --help
```

The next step is to make dependent parts of the software, then compile the software itself:

```
make depend
make
```

Before you install the software, this is a good time to test that it actually works. The **make** script provides a test routine that will verify the parts of LDAP you have chosen to install are working:

```
make test
```

The test routines may take some time: If anything unusual is discovered, you will be given plenty of information on the problem. When everything is working OK, you can install the software by switching to root and using the customary:

```
make install
```

The server software is now installed on your system and ready to run, but there are a few more tasks that you will have to perform before you can use it...

Post-install

Before you can start your server, you need to modify the *slapd* configuration file. The *slapd* daemon runs your actual LDAP server, and needs some specific information before it can run. You should find the file in /usr/local/etc/openldap/slapd.conf. The file has already been created, and contains lots of comments about the available options. Open it with your favourite text editor so you can make some changes.

The most important change to be made is to define the scope of this server. At the beginning of this tutorial, we mentioned the different domain hierarchies – it's time now to decide which convention you are going to use.

»

LDAP JARGON

DN – Distinguished Name
While there can be many entries for the same item, each must be able to be referenced at a single part of the hierarchy – the Distinguished Name effectively walks down this hierarchy to a specific, unique item.

LDAP – Lightweight Directory Access Protocol
A directory service that conforms to the LDAP standard.

Slapd
This is the server daemon that handles LDAP requests. It uses a backend database to actually store the data.

Slurpd
The replication server is for a distributed directory, shared across more than one server.

Referral
LDAP has the ability to redirect requests it cannot handle (*eg* out of scope searches) to another server. The referral address is defined in the config file.

ACKNOWLEDGEMENT
Many thanks to Rackspace Managed Hosting, who have donated hardware to help us complete this Server School tutorial series. For more info about Rackspace, turn to page 94.

SCHEMAS AND CLASSES

The schemas used in LDAP are like recipes for objects. They are refereed to by the objectClass entries in the LDIF – each class is defined in one of other of the schemas. The attributes available to each entry are defined by which classes you use, rather like defining an object in a high-level programming language.

Some of the schemas reference each other (*eg* to use inetOrgPerson you'll need to also load the cosine schema). You can find a lot more in-depth explanation of the classes and objects at http://ldap.akbkhome.com/index.php

« You'll find the sample database definition at the end of the file. The suffix line must be altered to reflect the distinguished name of your own LDAP directory.

You'll also find an entry below that, **rootdn**. This is a temporary entry specifying the manager of the directory (and rootpw, a few lines below, specifies the password). Make sure the rootdn is changed to reflect your directory tree. The directory normally stores information within it as to who is allowed manager access – but as we have yet to put any info in the directory, this is a get-around to allow us access. We'll be coming back to secure the directory after we have included some data.

You also need to specify how you want to index the directory. Indices are created to allow searching by different attributes, and a number of different modes are available. These may well not make much sense to you until you decide on a schema to use and work out what information is to be stored, but we'll go through the basics anyway.

The simplest index is an **eq** index, which simply matches entries exactly equal to the search term. Alternatives include **sub** which returns results with substring matches and **approx** which applies a kind of fuzzy filter. For example, suppose we had an entry with a common name attribute **cn= Nick Veitch**. An equality index would only return exact matches, but for some purposes – say, for example, an email client using LDAP to complete addresses – we may want a substring or fuzzy match so that **Veitch**, **Mick Veitch** and other searches were also able to find the right record.

Some index types aren't available for certain attributes though. It's always worth checking into the documentation if you have difficulties. A typical set of index lines in the config file might look like this:

```
Index default eq
Index cn,sn eq,sub,approx
```

There are a few other changes you'll want to make. One that springs to mind is to include schema definitions in the global settings. At the beginning of the config file, you'll find a line beginning **include** with the path to the core schema. We need to add a few more to this:

```
include     /usr/local/etc/openldap/schema/cosine.schema
include     /usr/local/etc/openldap/schema/inetorgperson.schema
include     /usr/local/etc/openldap/schema/openldap.schema

```

We'll cover the schemas briefly in a minute, for now just enter this as a magic spell.

There are a few more things to do with the config file, like look at access control. For the moment it is set up fairly loosely, but this is okay while we fill it with data – we'll change the settings when there is some live data to work with.

Setting up the directory

There are several ways of getting data into the directory, but in my opinion, the simplest way is to run it and use the client software to add data. To start the directory daemon, we need to run *slapd*. If you chose the default install path for the software, you'll find the executable in /usr/local/libexec/slapd.

There are a number of options for starting the server, which we'll go through briefly here

-d <level> Debugging level. There are many options here – run with **-d ?** to get a list.

-f <filename> Use a specific configuration file – defaults to /usr/local/etc/openldap/slapd.conf

LDIF EXAMPLE

```
#Example LDIF file
dn: dc=lxf,dc=net
dc: lxf
description: Linux Format
objectClass: dcObject
objectClass: organization
o: Linux Format org

#Entry for the DM
dn: cn=Manager, dc=lxf,dc=net
description: Directory
objectClass: organizationalRole
cn: Manager

# ou= people unit
dn: ou=people, dc=lxf,dc=net
ou: people
description: People group
objectClass: top
objectClass: organizationalUnit

#person entry
dn: uid=nveitch,ou=people, dc=lxf,dc=net
telephoneNumber: 2335
l: Bath
userPassword:{SSHA} U50F5ThTzzpZcubWUsr0H8GUPSNdzkgN
uid: nveitch
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgPerson
sn: Veitch
cn: Nick Veitch

#person entry
dn: uid=phudson,ou=people, dc=lxf,dc=net
telephoneNumber: 5038
l: Bath
userPassword:{SSHA} U50F5ThTzzpZcubWUsr0H8GUPSNdzkgN
uid: phudson
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgPerson
sn: Hudson
cn: Paul Hudson
```

-n <name> Specifies a name for the service for logging *etc*. Only really useful if you run multiple servers.

-h <hosts> Specify the listener configuration. This defaults to LDAP over TCP on port 389, but you can use **ldaps://** for listening over *SSL* (on the default port 636) or any other connection.

-u <user>: specify the user to run as

-g <group>: specify the group to run as

-r <directory> : specifies a runtime directory. The *slapd* daemon will **chroot** to this directory after opening listening ports (but BEFORE reading config files etc). This is useful for extra security.

For the moment we can run

```
/usr/local/libexec/slapd
```

to initialise the server. At this point, we should probably say that the correct way to stop **slapd** is:

```
kill -INT `cat /usr/local/var/slapd.pid`
```

At the moment, there is no data at all in there, not even the root DN. The easiest way to create data is in a text file, using the LDIF file format. Fire up a text editor (and here's a tip – the text editor *kate* understands the LDIF syntax) and enter the text.

Every entry must have a DN which attaches to the tree. The reference is one of the attributes of the entry. Permissible attributes in LDAP are defined by schemas. There are a number of schemas to suit all sorts of information, so though it is possible to create your own, it would be true to say that generally you would be best off with sticking to the methods provided. A sample LDIF file is shown in the *LDIF Example* box at the bottom of the facing page.

This LDIF sets up the basic structure of your directory. The first entry defines the root, and should be quite straightforward. The entry for Manager is just an entry that reflects the Manager details we specified in the config file. It isn't required initially to edit the information, but to be more secure we want to move control of the directory to an admin with a proper password etc. You can include it anywhere in the hierarchy you like, but leaving it at the top makes it easy to find.

The first bit of structure you come across is the section beginning **dn: ou=people,dc=lxf,dc=net**. This is an organisational unit, or a container for easier management of entries. A **people** container is fairly common, and you may want to have subcontainers for different job roles, locations or whatever kind of data you are managing.

The next entry, for a person, should make the structure even clearer. The person with **uid=nveitch** is defined, in relation to the ou=people container. This is a unique entry which specifies the data belonging to that object belongs to that particular part of the tree. If you find this confusing, try to think of it as a filesystem. Instead of **uid=nveitch,ou=people,dc=lxf,dc=net**, think of the **dn** as a filename and read it backwards:**net/lxf/people/nveitch**

The different attributes available to be defined relate to particular schemas. The schemas themselves are invoked for an object by the **objectClass:** definitions. Some schemas require certain attributes to be defined for the entry to be valid.

With the LDIF file created, and the directory still running, we can import the data into the directory using:

```
ldapadd -f ourdata.ldif -x -D “cn=Manager,dc=lxf,dc=net” -w
secret
```

substituting the filename and **dn** in an appropriate way for your configuration.

You should get a message letting you know if the update has been successful. LDAP is famed for being pernickety about tiny syntax details – you should make sure you leave a blank line between entries for instance.

At this stage, you might want to experiment by browsing the directory to make sure things are as you expect. It may be a good time to fire up your own LDAP client (or take a look at the box entitled *LDAP Utilities* at the bottom of the next page).

PHP AND LDAP

There are plenty of web-based tools to help you query and manage LDAP directories. Languages like Perl, Python and PHP all have good LDAP support, so it is relatively easy to write your own. We covered creating various types of PHP clients in the PHP tutorial back in *LXF52* – turn to page 97 to order your backissues.

Security

We said earlier on that once we had the database up and running, we'd come back and sort out the security aspects. The first place to do this is in the slapd.conf files, where we can alter the access control. There will probably be some commented out lines in the config file relating to access control. Leave these for now, we'll determine a local access policy by adding our controls beneath the index lines we added earlier.

Access controls start with the **access** directive naturally enough, and follow the form:

Access to <what> by <whom> <access-type>

The **<what>** refers to particular elements of the hierarchy, **<whom>** is the user, entity, range of IP addresses or whatever users we wish to apply this to, and the **<access-type>** is the type of access we are granting.

To start with the last element first, there is an ascending scale of six types:

- **none** – Refused access
- **auth** – Bind access, can access the database
- **compare** – Can compare given values
- **search** – Can apply search filters
- **read** – Access to read the search results
- **write** – The highest access, you can read and modify data

Only **none, auth, read** and **write** are commonly found, but these examples are fairly self-explanatory.

The **<what>** parameter determines the parts of the hierarchy this directive applies to. There are several options here, but we will only look at the more common ones (feel free to check out the information thai is provided in the *OpenLDAP* documentation if you want to get more complex).

The * wildcard character relates to the entire hierarchy. Alternatively, you can supply a **dn scope**, which can be **base**, **one**, **subtree** or **children**.

- **base** – the provided DN only
- **one** – the children of the provided DN only
- **subtree** – the entire hierarchy below and including the dn
- **children** – the hierarchy below – but not including – the supplied dn

The **<whom>** parameter can get a bit complex. There are a few special cases:

- * – everybody (including non-authenticated users)
- **anonymous** – specifically anonymous users
- **users** – all authenticated users
- **self** – the user associated with the target entry
- **dn** – users identified by the supplied **dn scope**. This works much the same way as the **<what>** parameter.

»

SLAPD.CONF EXAMPLE

```
# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.23.2.8
2003/05/24 23:19:14 kurt Exp $
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include    /usr/local/etc/openldap/schema/core.schema
include    /usr/local/etc/openldap/schema/cosine.schema
include    /usr/local/etc/openldap/schema/inetorgperson.schema
include    /usr/local/etc/openldap/schema/openldap.schema

# Define global ACLs to disable default read access.

pidfile     /usr/local/var/slapd.pid
argsfile    /usr/local/var/slapd.args

defaultaccess none

#
# if no access controls are present, the default policy is:
#    Allow read by all
#
# rootdn can always write!

#######################################
# ldbm database definitions
#######################################

database    bdb
suffix       "dc=lxf,dc=net"
rootdn       "cn=Manager,dc=lxf,dc=net"
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw        {SSHA}0gIEm65+LpxKkIA+50Z8AhIE6SZTsYu7
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.

directory    /usr/local/var/openldap-data

# Indices to maintain
index default eq
index cn,sn,uid pres,eq,approx,sub
index    objectClass    eq

#specific access for this directory
access to attr=userPassword
        by dn="cn=Manager,dc=lxf,dc=net" write
        by self write
        by * auth

access to *
    by dn="cn=Manager,dc=lxf,dc=net" write
    by self write
    by * read
```

« A typical set of access controls may be:

```
access to dn.children="ou=people,dc=lxf,dc=net"
     by self write
     by anonymous auth
     by * read
```

The order is important, only the first met condition is acted on. In this case, members of the **ou=people** can write data to their entries, **anonymous** have **auth** access, and everyone else can read. Because '**anonymous**' is matched in the list before the **by * read** line, anonymous users will not get read access. The order by which the access directives works in the same way, so it's useful to remember to deal with specific cases first, before implementing general controls.

In our example, we'll be using user passwords to determine access. Users will be able to view and change their own data, and view other entries, but not change them. Anonymous users will be able to read the entries, but not the password (or even the hash of the password).

When your directory has more information in it, you may wish to add further restrictions, like applying anonymous read access only to certain attributes, or only to certain organisational units.

Before we are finished, we need to say something about passwords. Cleartext passwords aren't a good idea. The *slapd* server supports all sorts of authentication protocols (including some which you may have specified at install time such as *Kerberos* and *SASL* methods). At the very least you should be using salted SHA or MD5 passwords.

The utility supplied with *OpenLDAP*, *slappasswd*, will generate a hash for you to enter. This is a fairly trivial operation for Perl or PHP too if you are creating your own admin interface, and many of the Open Source ones have similar functionality. You should certainly create a hash for the **rootpw** in the slapd.conf file. Run the *slappasswd* utility and having entered the password, you'll get output like this:

```
{SSHA}U50F5ThTzzpZcubWUsr0H8GUPSNdzkgN
```

Cut and paste this into the slapd.conf file.

For even better security, you should run LDAP over *SSL*. This will prevent even the hashed passwords being exposed on the network (which can be cracked given enough time and determination). There isn't really space here to describe this procedure, but if you want to see it covered in future, please write in and let us know.

Above is the final config for our example server.

Finally

There is plenty more to be done with LDAP. Apart from securing it even further, and perhaps implementing *Kerboros* or *SASL*, there's a question of what else you may want to integrate LDAP with. *Apache* is a good example, where LDAP can implement easy-to-manage fine-grained access policies, or there is just implementation as a standard NIS service. If you have any specific requirements, write in and we may cover them in a future tutorial. LXF

LDAP UTILITIES

As well as the utilities provided with *OpenLDAP*, there are a number of programs, scripts and web examples of administration clients. It's not surprising many of these are web-based, but that does mean there is cross-platform compatibility if you wish to manage the directory from Windows or a Mac platform.

A quick search of freshmeat will bring up quite a few responses for LDAP, but one of the best utilities we found was called simply *ldapbrowser*. It's a Java application and you can find it at www.iit.edu/~gawojar/ldap/

TAMING YOUR HARD DRIVE

Command-line: hdparm

Want your hard drive to be a screaming, power-sucking speed demon or a quiet, green angel? Richard Drummond has the answers.

Like many aspects of the of the Linux kernel, the IDE (ATA) driver is tunable. This is because, as the ATA specification has evolved over the years, new and faster transfer modes have been added and various optional features catered for, such as power and noise management, and thus your average ATA hard drive will support a variety of these modes and features. Ideally, the Linux IDE driver should set up your drive for optimum performances, but, depending on how your kernel is built and on your system's ATA adapter, it may sometimes get it wrong. Thus this 'tunability' is a good thing if you want to crank the last erg of performance from your drive.

The key to tuning the ATA driver's parameters is the shell command **hdparm**. This lets you query and modify various driver settings, get information about your drive and its capabilities, perform various tests and to enable or disable optional features. To do its job, **hdparm** must be run as root, and it shouldn't be used lightly. You shouldn't really be able to damage anything by misusing **hdparm**, but you may hang your system. In short, if you want to experiment, make sure you have no critical files open first.

The need for speed

For most users, the primary use for **hdparm** will be making sure your hard drive is operating as fast as it can be. So how do you do this? First of all you need to know a little about the transfer modes that the ATA specifications support (see table). Secondly, you should know what speed your hardware supports (note that both your drive and your ATA adapter are limiting factors). The best place to start is to do a speed test. This can be done like this:

```
hdparm -t /dev/hda
```

The **-t** switch tests read speed from the drive. /dev/hda it the the device node of the drive to query (here the primary drive on the master channel). I get with a fairly old ATA/100 chip set and a brand new ATA/133 drive – the result:

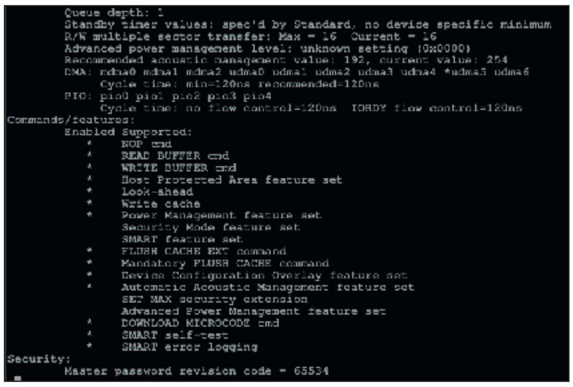
```
Timing buffered disk reads: 116MB in 3.02 seconds=
38.41MB/sec
```

If your system's not entirely prehistoric, you should get a comparable result (you should at least get double figures). Note that you'll never get anything like the theoretical maximum transfer speed. However, if you get a report of 1 or 2 MB/s, then your system is using a PIO transfer mode rather than a DMA mode (some distributions don't enable DMA by default, for safety reasons). That's bad for speed. To correct this, run:

```
hdparm -d1 -c1 -m16 /dev/hda
```

The **-d** switch enables or disables DMA (following it with **0** disables, **1** enables); the **-c** switch enables or disables 32-bit transfers to the ATA controller (not the drive, this still uses 16-bit transfers. However, all PCI chip sets should support 32-bit transfers). The **-m** switch specifies the maximum number of disk sectors to transfer in a single request (16 means 2^{16} or up to 65535 sectors. With pre-ATA-6 systems use **-m8** instead.).

Test your system thoroughly, and, if it still seems stable, add that **hdparm** command to your init sequence (after the filesystem checks). The script /etc/rc.boot will be suitable place for most systems.



```
        Queue depth: 1
        Standby timer values: spec'd by Standard, no device specific minimum
        R/W multiple sector transfer: Max = 16  Current = 16
        Advanced power management level: unknown setting (0x0000)
        Recommended acoustic management value: 192, current value: 254
        DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 *udma5 udma6
             Cycle time: min=120ns recommended=120ns
        PIO: pio0 pio1 pio2 pio3 pio4
             Cycle time: no flow control=120ns  IORDY flow control=120ns
Commands/features:
        Enabled Supported:
           *    NOP cmd
           *    READ BUFFER cmd
           *    WRITE BUFFER cmd
           *    Host Protected Area feature set
           *    Look-ahead
           *    Write cache
           *    Power Management feature set
                Security Mode feature set
                SMART feature set
           *    FLUSH CACHE EXT command
           *    Mandatory FLUSH CACHE command
           *    Device Configuration Overlay feature set
           *    Automatic Acoustic Management feature set
                SET MAX security extension
                Advanced Power Management feature set
           *    DOWNLOAD MICROCODE cmd
           *    SMART self-test
           *    SMART error logging
Security:
        Master password revision code = 65534
```

hdparm gives a wealth of information on your hard drive's configuration and features.

Noise and power

The other aspect of **hdparm** is enabling optional features such as power and acoustic management. If your drive supports these, you can configure your drive to power down after a certain period of inactivity or to operate more quietly.

First check what features your drive supports with the **-I** switch. You should see something like the screenshot above. If your drive supports power management, you can tell it to enter standby after a specified period of inactivity. For example:

```
hdparm -S20 /dev/hda
```

This will make the drive 'spin down' after 100 seconds of inactivity. The number following the **-S** switch will set the spin-down time in multiples of 5 seconds. You can force a drive into standby mode with the **-y** switch and put it to sleep completely with the **-Y** switch.

Another optional feature is 'acoustic management.' If supported, this controls the amount of noise your drive creates – at the expense of speed. This is controlled **-M** switch followed by a numeric parameter in the range from 128 to 254, where the former means 'quiet', the latter 'fast'. **hdparm** supports many more commands and options. See the man page for more details. LXF

ULTRA DMA MODES

What those numbers mean.

UDMA mode	Marketing name	Maximum transfer rate	ATA standard (1)
0	N/A	16.7 MB/s (2)	ATA/ATAPI-4
1	N/A	25 MB/s (2)	ATA/ATAPI-4
2	Ultra-ATA/33	33 MB/s	ATA/ATAPI-4
3	N/A	44.4 MB/s (2,3)	ATA/ATAPI-5
4	Ultra-ATA/66	66.7 MB/s (3)	ATA/ATAPI-5
5	Ultra-ATA/100	100 MB/s (3)	ATA/ATAPI-6
6	Ultra-ATA/133	133 MB/s (3)	ATA/ATAPI-7 (4)

1 See www.t13.org/
2 Never supported by production drives
3 Mode 3 and above require an 80-wire cable
4 Not formally released yet.

USING KDEVELOP

KDE Development

PART 2 **In an effort to show his Aunt Kath his photos, Jono Bacon begins a series on developing KDE applications.**

On the CD and DVD

Welcome to a brand spanking new series that will be focussing on the often mysterious but intriguing art of KDE development. Since the inception of KDE many moons ago, the project has continued to develop consistent releases that are not only admired for their ease of use and visual quality, but also ease of use from a developmental perspective.

There is no doubt that KDE is a large and expansive programming environment, and there are many twists and turns in the maze of technologies, protocols, standards and practices. As any established programmer will know, one thing is learning how to program a particular system and another thing is programming it the *right* way.

The aim of this series is to take us from scratch and build a complete and functional application. While we move further and further into the details of creating our project, the techniques that are common on KDE development will not only be learned, but applied. The aim of the game here is not to dwell on theory too much – we want to get down and do some practical coding. This has always been the best way of learning, and this is the technique we will be using over the coming months.

This series is very much an interactive affair. I would love to hear your comments, suggestions, criticisms and other views. Remember that Linux, KDE and the community that maintain it is a collaborative community, and we need to point out issues and make suggestions to get the most out of it. This is no different for this series. We have the space and the content, but your views, opinions and suggestions can make this space even more valuable; please send them onto me at **jono@jonobacon.org**.

In this series, I will be assuming that readers have a certain amount of knowledge. Firstly, I will assume that you know C++ to a reasonably competent level. I am certainly not assuming any kind of cutting-edge C++ skills, but you should know how to create classes, instantiate objects and use methods. In addition to this, I expect that you know what KDE is; this generally helps! Without further ado, let us begin.

Designing our interface

With any kind of application development, it is a good idea to have a clear idea of how your application will be structured before you write it. This not only involves an idea of what it will do, but also how it will be visually designed. We need to decide what kind of interface we are going to present to our users and how the interface will ensure that the application is simple and easy to use.

The design that we are going to use in *LXFGallery* is shown in **Fig1**. As you can see, we have three major areas in the application. The top left album area will be a list of our current albums that we have saved. We will also probably throw into the mix a little button to create a new album, to make it a more straightforward process than clicking through menus.

The section below the albums area will display the thumbnail images from the selected album. This way, we can click on an album and instantly see images directly below it. If we then click on an image, we will see it displayed in the large area to the right.

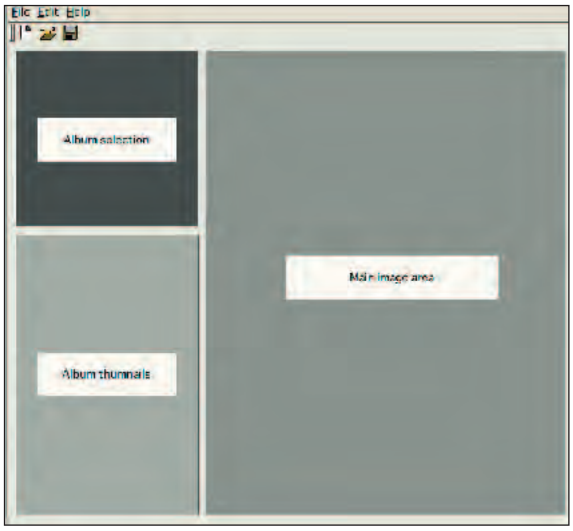


Fig1 **Our general interface design for *LXFGallery*.**

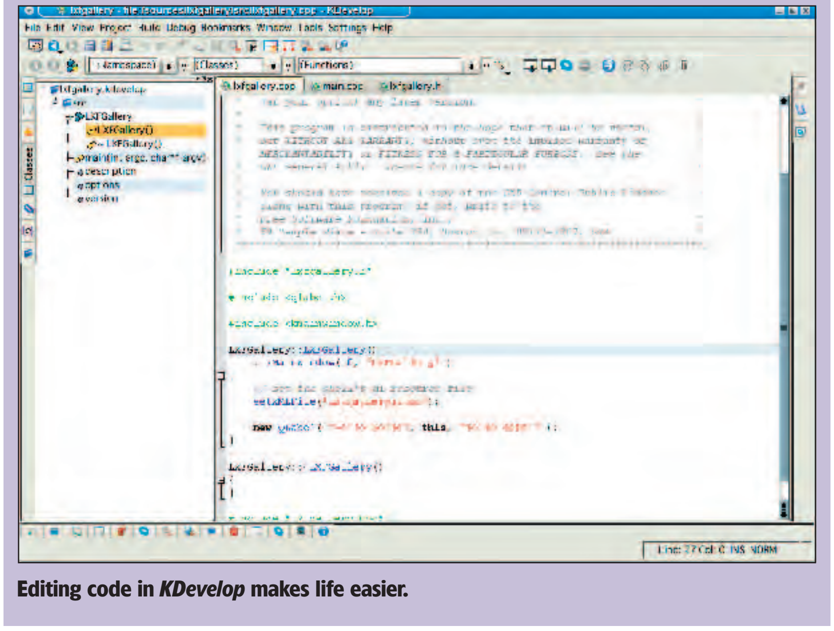
THE KDE TOOLBOX

Powerful and flexible development

When you open up the bonnet of the KDE desktop, there is a *lot* on offer for the developer. KDE is not just a framework to create graphical applications, but a complete subsystem to deal with most kinds of information and problems that you will come across. This framework is largely built upon the powerful *Qt* toolkit from Trolltech. This C++-based graphical development system gives you the ability to create applications easily and includes a number of convenience facilities that make common programming tasks much easier.

In this series we are going to be using the *KDevelop* IDE to write our software with. The reason for this is simple. *KDevelop* is a powerful and flexible tool that not only simplifies our development, but it lets us plug in additional tools that can help us refine and stabilise our projects. You can think of *KDevelop* is a springboard for everything in our development, and we will use it to pull the different strings to craft our project.

To get started with KDE development you will need a number of tools installed on your system. First, you will need the *Qt* toolkit. You can get this from www.trolltech.com/ or with your distribution. You will also need the KDE system and importantly, the KDE programming libraries. This is typically available within distributions as a *kdelibs-dev* package. If you simply have the KDE interface on your machine this will not be enough. You need to programming header files to create your applications and these are present in the *kdelib-dev* packages.


lxfgallery - file:/sources/lxfgallery/src/lxfgallery.cpp - KDevelop
File Edit View Project Build Debug Bookmarks Window Tools Settings Help
(Classes)
(Functions)
lxfgallery.cpp
main.cpp
lxfgallery.h
lxfgallery.kdevelop
LXFGallery
LXFGallery()
~LXFGallery()
main(int argc, char** argv)
description
options
version
Classes
Line: 27 Col: 0 INS NORM

Editing code in *KDevelop* makes life easier.

This main area will be where the image will be displayed in its large size, and we will also display caption information alongside it.

Although these three areas have been explained, we must not forget the other parts of the interface. Right at the top of the window, we have our menus and toolbar. These menus will be used and added to as our little project matures, and the toolbar will have the most commonly used buttons added to it. The final area is the status bar at the bottom of the window. This bar will display short messages and details about how the application is functioning. Although a small detail in the grand scheme of the project, the status bar is key to ensuring that your user always knows what is going on – this will establish confidence with your audience.

At this point we have a general idea of what our project will do and how it will look. Don't worry too much about being aware of how all this fits together at the moment: much of this will become clearer as we write code, compile it and change it.

Creating code

With a loose idea of the direction that we are going with *LXFGallery*, we will jump right in and create some code. As I said earlier, the best way to learn is by doing, so we will generate some code and then discuss how it works. Luckily, much of this automated with *KDevelop*.

Load up *KDevelop* and select Project>New Project. In the list of projects you can select C++>KDE>Simple KDE Application. Type in the name of your application as 'LXFGallery' (making sure that you type this in the right case, as it will be used as your class name). You can then keep clicking on Next until your project is finished. I will not be discussing version control in this series, but if you want to use the version control features in *KDevelop*, feel free to do so.

The code will now be generated and if you click on Build> Build Project (or press **F8**) you will be asked if you want to create Makefiles and build your project. Select yes and your code will be built. There should be no errors in the code and you can then run the program with Build>Execute Program (**Shift-F9**). You will then be presented with a window with the words "Hello World!" displayed in it as shown in **Fig2**. If for some reason you encountered some errors in this compilation process, then it is likely that some required software is not installed on your system.

When you generate the code we have a number of files generated for us. These include:

- **main.cpp** – contains our **main()** function that starts the application
- **lxfgallery.h** – header file for the **LXFGallery** class
- **xfgallery.cpp** – implementation file for the **LXFGallery** class

We also have a load of other files generated for us, but these three files are the main source code files that we are interested in, and they reside in the *src* directory. The files basically supply a main **LXFGallery** class that is used to create an object in the **main()** function in main.cpp. We will first look at this class and see how it works. We will then move on to explore the **main()** function. I will reference specific parts of these files so as not to reproduce all of the comments.

First, let us take a glance at lxfgallery.h. We first see some include files:

```
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <kmainwindow.h>
```

We first use an **#ifdef** to see if config.h has been defined already. This is not overly important at this stage in our development, but we do need to include kmainwindow.h. This file contains a class called **KMainWindow** that can be used to provide typical office type application functionality. This kind of functionality covers toolbars, status bars and the general application. We are including this file here as we will inherit from it with our **LXFGallery** class.

Here is our main class:

```
class LXFGallery : public KMainWindow
{
  Q_OBJECT
  public:
    LXFGallery();

  virtual ~LXFGallery();
};
```

Anyone who has done any kind of programming before will be familiar with this kind of code. We create a **LXFGallery** class and inherit from a **KMainWindow**. What might look a little unusual is the **Q_OBJECT** part of this code. This is a special macro that is used by *Qt* to enable interaction with the application. We will not

»

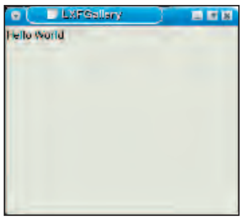


Fig2 **Whatever development platform or language you choose to try out, a simple program like this is likely to be your first step.**

INTRODUCING LXFGALLERY

Programming our project application

To make this series more interesting than just examining theory, we are going to develop and build our very own gallery application. The aim of this project is to let us manage our digital camera photos and put them into albums for viewing. We will also be including a few particular special features that will give *LXFGallery* some real value to the user.

To give you an idea of the kind of features we will include in *LXFGallery*, cast your eye over this very basic initial list of ideas:

- **Allow the user to create a new album and add pictures to it**
- **Provide a means for every image to have a caption associated with it so the user can jot down memories or provide an explanation of the picture**
- **Save albums to the disk**
- **Create a HTML album so that you can publish your pictures to the web**

This is only a simple list of features, but it already gives us a lot to get started with. As ever, we welcome requests for clarification, suggestions of subjects to be included, so send your feedback – positive or negative – to either jono@jonobacon.org or linuxformat@futurenet.co.uk with *"KDevelop Tutorial series"* as the subject-line.

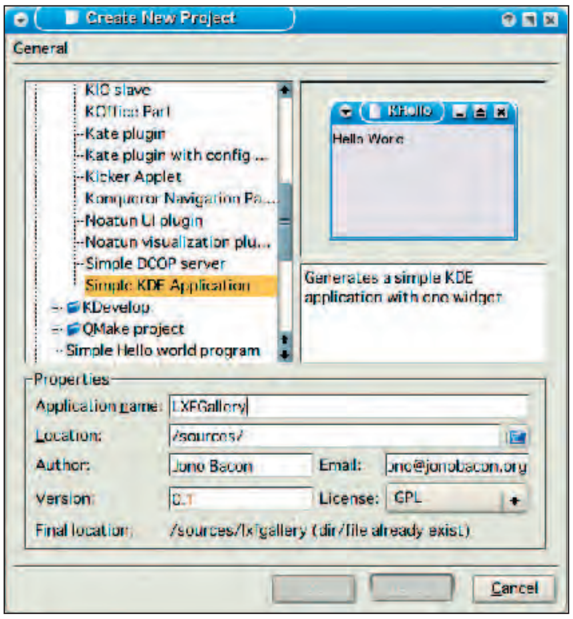
Before we begin with the knotty task of development, some forethought is needed: we need to consider how our application is going to be visually designed – see the section under the heading *Designing Our Interface* on the previous pages for more.

« be covering **Q_OBJECT** in this issue, so just happily ignore it for now! You will notice also that only the class definition is shown in the header file. This is fairly common and we are essentially saying that this class will have the **LXFGallery()** constructor (this function is run when the class is used to create an object) and the **~LXFGallery** destructor (this function is run when the object is destroyed) in the class implementation.

The actual code for these functions is in the lxfgallery.cpp file. In this file we first include the relevant header files:

```
#include "LXFGallery.h"

#include <qlabel.h>

#include <kmainwindow.h>
#include <klocale.h>
```

We first include our class header file lxfgallery.h and then we begin to include some normal API header files. The first, qlabel.h, is used to create the "Hello World!" text label in our window. This is a *Qt* class as it begins with a **q**. KDE classes rather unsurprisingly begin with **k** in the filename and we include two here, kmainwindow.h and klocale.h. We have already covered kmainwindow.h but klocale.h is new to us. This class allows us to support different languages and locales in our application. Although this is encouraged, we will only be focussing on English in this series so you can remove the klocale.h file.



Generates a simple KDE application with one widget

Application name: LXFGallery
Location: /sources/
Author: Jono Bacon
Email: jono@jonobacon.org
Version: 0.1
License: GPL
Final location: /sources/lxfgallery (dir/file already exist)

We can create a new project quickly and easily with the *KDevelop* New Project wizard.

Next we have our main constructor. This constructor inherits from **KMainWindow** and is passed two arguments:

```
LXFGallery::LXFGallery() : KMainWindow( 0, "LXFGallery" )
{
```

The first is the parent for the widget. In this application it is not set (hence **0**), and when there is no parent for a **KMainWindow**, it makes the **KMainWindow** widget toplevel. In other words, if you don't set the parent for the **KMainWindow** inherited class, the widget will be displayed in its own window. This means that we will display our *LXFGallery* widget that is a **KMainWidget** behind the scenes as the main interface. The second argument is a name for the widget. This name is typically used for debugging purposes.

The actual content of the constructor is fairly simple:

```
setXMLFile("lxfgalleryui.rc");
new QLabel( "Hello World", this, "hello label" );
```

The **setXMLFile** method is something that we will be covering in the next issue that is part of the entire KDE XML user interface system. The **QLabel** line then creates a text label in our window. This class has three arguments. The first argument is the actual visible text that appears, the second is the parent (we are using the C++ **this** to specify the current class's object) and finally we have a name for the widget that can be used for debugging purposes.

After the empty destructor function, we then have this line:

```
#include "lxfgallery.moc"
```

This line includes a special **.moc** file into our code. MOC is the Meta Object Compiler that is included with *Qt*, and we will be discussing this is detail when we look at the user interaction parts of *Qt*. You can leave this line in here for now and ignore it. Don't worry, code doesn't take things like this personally!

Our final file to look at is main.cpp. This file contains our starting **main()** function and this is where the whole show begins. We will be focussing on the **main()** function here, and the very first part of this function sets some details about the application:

```
KAboutData about("lxfgallery", I18N_NOOP("LXFGallery"),
   version, description, KAboutData::License_GPL, "(C) 2004
Jono Bacon", 0, 0,
   "jono@jonobacon.org");
about.addAuthor( "Jono Bacon", 0, "jono@jonobacon.org" );
```

Here we use the **KAboutData** class to create an object that stores some information about the class. This information is typically used in KDE in cases where this information is needed by the user or by KDE. The arguments that are passed to this class set the following information (from left to right):

- **Application name**
- **Program name**
- **Version**
- **Description**
- **License type**
- **Copyright statement**
- **Additional text**
- **Homepage**
- **Email address**

You should also note how we have used the previous static definitions of **version** and **description** from the top of main.cpp in this **KAboutData** object. After this object is created we then use the **addAuthor** method to add the first author to the application. This method is useful to easily add additional authors to the application's about details.

The next few lines handle the command line arguments that can be passed to the application when it starts:

```
KCmdLineArgs::init(argc, argv, &about);
KCmdLineArgs::addCmdLineOptions( options );
```

We will not be handling these arguments at the moment so we will ignore these lines for now. The next line is however, fundamentally important:

```
KApplication app;
```

Every KDE application must have a **KApplication** object. This object performs the lowest level donkey work that you generally don't need to be aware of. There are also some additional features in **KApplication** that can be used in certain situations. The main thing is to always include this line or you won't get anywhere with your KDE development!

The next line is where the fun begins:

```
LXFGallery *mainWin = 0;
```

Here we create an object definition of the **LXFGallery** class that we were discussing earlier. We create this object as a pointer and call it **mainWin**. We next open up an if/else statement that is used to determine session management. The first part checks if a session was open:

```
if (app.isRestored())
{
   RESTORE(LXFGallery);
}
```

Here we use the **isRestored()** method to determine if a session was available with the app (**KApplication**) object. If this is the case, we use the **RESTORE()** method to restore our application. If this is not the case, we then enter the else part of the condition. In this block of code, we perform the typical steps that we would run if we did not use any form of session management at all. We first need to create our **LXFGallery** object from our definition earlier:

```
mainWin = new LXFGallery();
```

When this object is created the **LXFGallery** constructor will then set off running. In this constructor we created the main text label that is displayed in the application. We next need to set our main widget:

```
app.setMainWidget( mainWin );
```

Generally, every app needs to use a main widget. This widget fills the main window area of the application and the widget that you set this to can be a container of many other widgets. Here we are setting this main widget to our **mainWin LXFGallery** object. With the main widget set we then need to actually show it:

```
mainWin->show();
```

A lot of new KDE/*Qt* developers are often surprised by having to actually display your graphical widget with **show()**; surely this is pointless in a graphical toolkit such as the *Qt* and KDE libraries? Well, no. *Qt* and the KDE libraries do actually have a number of facilities for dealing with non-graphical aspects of programming such as data structures, files, networking and more. It is in fact possible to write a comprehensive non-graphical KDE/*Qt* application, believe it or not. Due to these non-graphical elements, we need to specifically show the graphical elements. You only need to do this occasionally however.

After we have left our if/else condition, we then run our final line of code before the application begins:

```
return app.exec();
```

This line allows the application to begin responding to input from buttons, widgets and other interactive elements. This is rather redundant here as we don't actually have any interactive elements. Have faith, though – we will get fully interactive in coming issues.

Take a deep breath

We have just covered a lot of code and not all of it will be clear at the moment. Don't worry however; things will be become clearer as we continue in the series. For now, take a few minutes to catch your breath. To make this information a little easier digest, we will cover the overall process of getting the *LXFGallery* project on the screen from start to finish:

- We first enter the main.cpp file. Here we include some header files, one of which is lxfgallery.h that makes us aware of the **LXFGallery** class. This class in turn inherits **KMainWindow**.
- Next we enter the **main()** function and set some details about our application such as who wrote it and the license.
- Next we create a **KApplication** object. We also define a pointer to a **LXFGallery** object (these two objects are not related, by the way).
- We then go on to check if our **KApplication** object has a session open. If it does, we **RESTORE()** it. If it doesn't, we create our **LXFGallery** object, set it as the main **KApplication** widget, and then display it.
- Finally, we instruct the application that it can go off and handle its own input now. This will also become clearer in future issues.

NEXT MONTH

We have covered a lot in this issue. We recommend that you take some time to read this article through a few times and let the code sink in. Don't try to understand all of it, as you probably won't at this point. The main thing is that we have taken our first step and discussed the bulk of our application. Much of this code is present in every KDE application and never needs to be looked at again.

Next issue, we will be continuing our quest for KDE development and begin by developing the *LXFGallery* interface. Stay tuned, folks...

SOFTWARE REQUIREMENTS

Properly equip your PC for KDE development

These are the specific software requirements for *KDevelop* that you need to follow this series:

- **KDE (3.0.2+)**
  www.kde.org
- ***Qt* (3.0.5+)**
  www.trolltech.com/products/qt
- ***g++* (2.95.3 + or compatible)**
  gcc.gnu.org
- ***GNU make***
  www.gnu.org/software/make
- **Perl (5.004+)**
  www.perl.com
- ***autoconf* (2.52+)**
  www.gnu.org/software/autoconf
- ***automake* (1.6+)**
  www.gnu.org/software/automake
- ***Flex* (2.5.4+)**
  www.gnu.org/software/flex
- ***Berkley DB* (3.0–4.1)**
  www.sleepycat.com

In addition to these essential requirements above, the following tools are optional:

- ***ht://Dig* (3.1.6+)**
  www.htdig.org
  A library used for searching for text.
- ***Valgrind***
  http://valgrind.kde.org/
  A tool for finding and fixing memory leaks. Memory leaks are important to avoid when create KDE applications in C++.
- ***GDB* (5.0+)**
  www.gnu.org/software/gdb
  A full featured debugger for finding and squashing those bugs. *KDevelop* integrates *GDB* within the *KDevelop* environment.
- ***CVS* (1.10.6+)**
  http//www.cvshome.org
  The most popular source control system, *CVS* is commonly used when a number of developers are working on one codebase.
- ***Perforce* (2003.1 +)**
  www.perforce.com/perforce/products.html
  Version control and configuration management.
- ***CTags* (5.x +)**
  http://ctags.sourceforge.net
  Special tags for easing the editing and development of code. This is built right into the *KDevelop* editor.
- ***Doxygen* (1.3.4 +)**
  www.doxygen.org
  Automatic generation of programming APIs and documentation.
- ***dot* (1.8.7 +)**
  www.graphviz.org
  Generation of graphical class diagrams. This is a highly recommended tool, as you will find that the class viewer is very useful when programming KDE applications.

Most of these tools are available in RPM and Debian package formats, and source code tarballs. If you are using a distribution that allows you to download packages from the Internet and install them automatically, you should have few problems with installing *KDevelop* and its optional extras.

Both CD and DVD readers will find this whole selection on this month's coverdiscs. The specific software from the first part of the list will be included in some later issues, space permitting.

GAME PROGRAMMING WITH SDL

Music – the food of love...

On the CD and DVD

21st Century games have stunning graphics, advanced artificial intelligence, and specially written soundtracks. Well, *one* out of three ain't bad, exclaims Paul Hudson...

Sometimes I completely forget to appreciate quite how lucky I am to have Nick Veitch as my boss. As the Editor of *Linux Format*, it's Nick's job to steer us in what usually ends up being something within shouting distance of the right direction, give informed and constructive comments on editorial copy, and attend various PR lunches at swish restaurants (it's a hard job, but someone has to do it). However, many of you will remember Nick from his earlier days working for Future Publishing as the Editor of *Amiga Format*, back in the days when having a 24-pin dot matrix printer was considered a luxury.

One of the benefits to this is that Nick spent years playing various Amiga games, which means that no matter how poor our SDL game is, chances are Nick's seen worse [don't be so sure – Ed], which in turn means I get free rein to move at my own pace in developing *Trout Wars*. Although our graphics still pretty much suck, we're going to leave them as-is right now and move onto the topic of sound and music – the art of making people ignore your hard work.

"Did you say *ignore*?" Yup, ignore. Think about it: when you go to the movies, the last thing you want to do is focus on the soundtrack, because you're there to see the flick. The best movies have very high-quality soundtracks that work so well you barely notice they are there. That's not to say they don't influence you – think of the *Lord of the Rings* theme, or the music from *Gladiator*. The audio should enhance your feelings and back up what's on screen, without stealing the show. It's a precarious balance, which is why good movie score composers are so highly valued.

In computer games, the situation is no different: the music should help to add depth and emotion to the game. Of course, unless you're Blizzard (if you've never heard the impressive soundtrack to *Diablo II*, you're missing out) you probably don't have the resources to have your symphony played by the London Philharmonic, but that's no excuse not to give our *Trout Wars* project at least *some* effort on the ear-candy front.

This issue we're going to be looking at how we can load music and sound files into our game. We'll also be examining how to make the game background look more interesting by putting scrolling stars in there.

Music response

First we're going to make some music play as our player moves around the screen. Ideally it should loop – start from the beginning as soon as it ends – thus ensuring continuous music. If you want to be really fancy, you could implement a playlist of sorts, but for now we'll stick with just the one song! In the future (probably about three issues away) we'll be implementing a user interface for the game that will need its own soundtrack, so I've written the code here to load two pieces of music – one for in-game, and one for the menu system.

All our audio will be handled through the *SDL_mixer* library that you should have installed as a result of following the instructions in the first tutorial in this series back in *LXF52*. If not, make sure that you grab it before you continue – we're going to need it. To enable *SDL_mixer* functionality, we need to make three simple changes – one in TroutWars.h and two in TroutWars.cpp.

In TroutWars.h, just after the line to include **SDL.h**, add this following line:

SUPPORT SDL

SDL is wholly supported by funding from the community, so why donate just a little money to help pay for development costs? The SDL homepage is at www.libsdl.org and there are instructions online for how to donate.

```
#include <SDL/SDL_mixer.h>
```

The first change in TroutWars.cpp is in the constructor **CTWGame()**. You should already have this line in there:

```
 if ( SDL_Init(SDL_INIT_AUDIO|SDL_INIT_VIDEO|SDL_INIT_
TIMER) < 0 ) {
```

That's where the audio is initialised, along with graphics and timers. However, although audio is initialised we still need to open the audio system, and that's done with this next line of code. Put this after the call to **atexit()**:

```
Mix_OpenAudio(44100, AUDIO_S16SYS, 2, 2048);
```

The first parameter there is the frequency of wave files to be played – 44.1 KHz is CD-quality, which is what most modern games use. If you want more backwards compatibility – or are eager to save space – you might want to use 22050, as it works better with older computers. The second parameter is the format of the sounds that will be played, with **AUDIO_S16SYS** meaning "Signed, 16-bit samples, in system byte order". This is recommended, and you should leave it as-is unless you have specific reasons. The last two are the number of channels you want, and the chunk size – again, you should leave them alone unless you really want to tweak things. The larger the chunk size is, the more likely the sound is to lag, but if you make it too small, the more processor-intensive it is.

At this point, we now have an open sound channel and can start playing our music. However, as we're being careful to avoid memory leaks, we also need to close the audio system once we've opened it. This is accomplished by adding this line into the destructor, ~**CTWGame()**:

```
Mix_CloseAudio();
```

To handle the music we'll be using three **#define**s – one to store the number of songs, and two more to store when they should be played. Put these three lines after the **#define**s for **SCREEN_WIDTH** and **SCREEN_HEIGHT** in TroutWars.h:

```
#define MUSIC_NUMSONGS 2
#define MUSIC_INTRO 0
#define MUSIC_INGAME 1
```

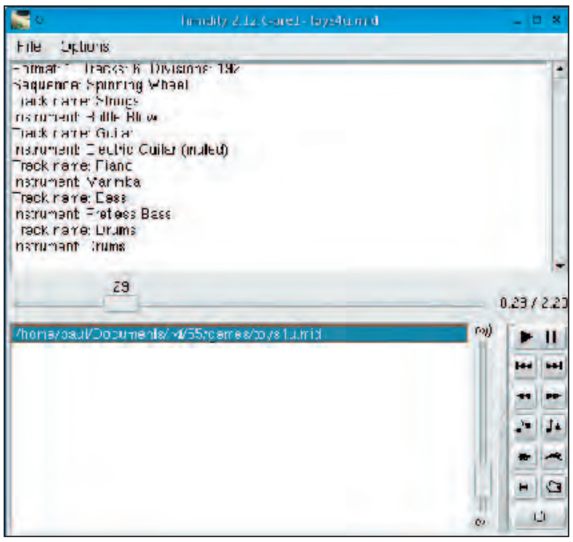
I'll explain how they work momentarily – first let me explain where the music is actually stored. SDL has a special structure for music resources called *Mix_Music*, and also has a variety of functions for loading and playing samples in this format. The three **#define**s above are there so we can create an array of *Mix_Music* structures, and use these **#define**s to index into the array to get the music we want. Put this line into TroutWars.h, inside the **CTWGame** class, under **sfcTxtMain**:

```
Mix_Music** Music;
```

Yes, that's a pointer pointer, but let me explain before you burst into tears. Pointers are best practice, as everyone knows, so the very least you should have is **Mix_Music* Music**. However, as we want to have an array of these, and arrays are just pointers to the first element, we actually end up with a pointer to a pointer – the first pointer points to the first element in the array, which itself is a pointer to a **Mix_Music** structure. Now, with that in mind, it should be more apparent why the **#define**s are so helpful – once everything is loaded we'll be able to access the in-game music like this:

```
Music[MUSIC_INGAME].
```

Actually loading the music is quite simple, in the same way that loading bitmaps is easy – SDL has special functions to handle the process, which means we can take a hands-off approach to it all. So, to load the music for our game, add these three lines to the end **CTWGame()** constructor in TroutWars.cpp:


TiMidity 2.12.0-pre1 - toys4u.mid
File
Options
29
0:23 / 2:23
/home/paul/Documents/lxf/55/games/toys4u.mid

```
Format: 1 Tracks: 6 Divisions: 192
Sequence: Spinning Wheel
Track name: Strings
Instrument: Bottle Blow
Track name: Guitar
Instrument: Electric Guitar (muted)
Track name: Piano
Instrument: Marimba
Track name: Bass
Instrument: Fretless Bass
Track name: Drums
Instrument: Drums
```

If MIDI playback sounds poor on your system, you could try running it through a program such as *TiMidity*...

```
Music = new Mix_Music*[MUSIC_NUMSONGS];
Music[MUSIC_INTRO] = Mix_LoadMUS("toys4u.mid");
Music[MUSIC_INGAME] = Mix_LoadMUS("sir_real.mid");
```

The first line there creates the array of **Mix_Music** structures. Literally, it says, "*Music* is a new pointer to an array of **Mix_Music** structures with **MUSIC_NUMSONGS** elements", which is just what we want. The next two lines use the **Mix_LoadMUS()** function, which takes a filename as its only parameter, loads the file, and returns the **Mix_Music** object with the music in. Note that I've used the **#define**s to position the music files in exact places, as planned. The files toys4u.mid and sir_real.mid are by me, and you're free to distribute them under the terms of the GPL. If you're talented musically and are willing to GPL some of your work, I'd be happy to consider a better soundtrack! These two are good enough for now, though, and they are on your coverdiscs this month.

As we've loaded the music we need to unload it, so add these lines to the ~**CTWGame()** destructor, before the call to **Mix_CloseAudio()**:

```
for (int i = 0; i < MUSIC_NUMSONGS; ++i) {
    Mix_FreeMusic(Music[i]);
}

delete[] Music;
```

All that remains now is to play the music, which, like everything else so far, is a walk in the park. Add this code to your **Play()** function, before the main while loop:

```
Mix_PlayMusic(Music[MUSIC_INGAME], -1);
```

Mix_PlayMusic takes two parameters – the **Mix_Music** to play, and the number of times it should loop. Specifying **-1** as the second parameter will make it loop forever, which is perfect for background music. Now compile it, this time being sure to add *-lSDL_mixer* to the **g++** call so that it links in the *SDL_mixer* library, and run the executable. All being well it should play the music and loop it indefinitely.

There's really no way anyone could describe playing back music as difficult, and there's some good news – **Mix_LoadMUS()** is able to load other file formats beyond plain old MIDI files. Depending on what you have installed, try loading Ogg files or various other formats. If you experience problems, make sure you have *libvorbis* installed and any devel packages. Beyond that, there's even better news – loading sound files is just as easy as loading music, so let's crack on... »

OUR IMAGE SOURCE...

Although the source code is GPLed and therefore free for you to use in any shape or form you like, some of images and sounds used in our game were borrowed from *The Games Factory*, a Windows-based game development tool from Clickteam (www.clickteam.com). Clickteam has also created *Multimedia Fusion* and *Jamagic*, which are much more advanced game development programs, both of which are currently being ported to Linux as you read this. Clickteam very generously agreed to place the resources used in this game under the GPL as well, which means you are free to use them as you please, so long as you mention that they are from Clickteam.com.

Sound off!

The key difference between sound effects and music is that music occupies a space by itself, whereas sounds mix together. Consider again the *Lord of the Rings* – when the impressive 'Helm's Deep' music is playing in the background, you hear it and only it. You *don't* hear the Hobbit theme or any other music – just that one piece. Compare that to the sound effects, where you might have three different sword sounds audible, some Uruk-Hai shouting, a horse galloping, *etc*, all mixed together.

Things are no different in games. As you've seen, playing music is as simple as selecting a piece and telling SDL the number of times you'd like it looped. Playing a sound effect is roughly the same, except you also need to tell it what channel you'd like it played on. As you're only allowed one sound effect playing per channel at a given time, you need to change the channels your sounds play on to make them mix together.

Again, SDL can do all this for us, which makes sounds straightforward. Sound is handled using the structure **Mix_Chunk**, so add this line into TroutWars.h beneath the declaration for Music:

```
Mix_Chunk* mixSound;
```

Naturally we'll want to have more than one type of sound playing in the final game, but this is just proof of concept right now to get you started with sound effects.

To load our sound effect, we need to call **Mix_LoadWAV()**, which, predictably, takes a sound file as its only parameter and returns a loaded sound ready for playing. To free it up once the game is finished, we need to call **Mix_FreeChunk()** and pass in the value we got back from **Mix_LoadWAV()**. In code form, here's how that looks. Add this line to the **CTWGame()** constructor in TroutWars.cpp, after the music is loaded:

```
mixSound = Mix_LoadWAV("laser1.wav");
```

The **laser1.wav** file is included on your coverdiscs, and was released by ClickTeam under GPL. Then add this line to the ~**CTWGame()** destructor in TroutWars.cpp, before the call to **Mix_CloseAudio()**:

```
Mix_FreeChunk(mixSound);
```

With our sound loaded, we can now go ahead and play it back whenever we want to. Once we have sound-making events in the game, we'll want to tie in the playing of sounds to these

STARRY-EYED GAMING...

Tarting up the *Trout Wars* background

So far the background for our game is a plain black one, which is pretty dull. What we're going to do is liven it up with a scrolling star field, made up of independently moving stars. That last part is quite important: if the stars don't move by themselves, you get a fixed star field which will look almost as unattractive as plain black!

To create the stars, we need two new classes: **CStarfield** will handle the drawing and movement of all the stars, and **CStar** will represent one star. Each star will need an X and Y co-ordinate, a speed variable so we know how fast to move it, and also a type variable so that we can have multiple star types. The **CStarfield** class will need to store an array of all the stars (for ease, this will be an STL vector, which is essentially a dynamic array), and an **SDL_Surface** for each of the star types. Here's how that looks in TroutWars.h (add it just below the line that says **class CTWGame;**

```
class CStar {
public:
int speed;
int xpos;
int ypos;
    int type;
};

class CStarfield {
public:
    std::vector<CStar*> stars;
    SDL_Surface** sfcStars;
    CTWGame* game;

CStarfield(CTWGame* localgame);
    ~CStarfield();
    void Draw();
    void Move();
};
```

If you've never used the STL before, the definition of **stars** in **CStarfield** might look scary. What it says is that we need a vector (the std:: part just says what *kind* of vector) that contains **CStar*** objects – that we want an array of pointers to stars. Note that the **CStarfield** class also has **Draw**() and **Move**() functions to handle, well, drawing and moving. In order to use vectors, we need to add this line after the line **#include <time.h>**:

```
#include <vector>
```

We also need to give our **CTWGame** class an instance of the CStarfield class, as each game needs a star field. Amend the start of **CTWGame** class definition to this:

```
class CTWGame {
public:
  SDL_Surface* sfcScreen;
  CPlayer* Player;
  CStarfield* Starfield;
```

As with remembering the names of music files, we'll also be using #defines to remember the three types of star: dark, medium, and bright. Add these four lines before the **#define** for MUSIC_NUMSONGS:

```
#define STAR_DARK 0
#define STAR_MEDIUM 1
#define STAR_BRIGHT 2
#define SCREEN_HEIGHT_TRIMMED SCREEN_HEIGHT - 5
```

The fourth **#define** is in there so that we don't create stars too near to the bottom of the screen – they are barely visible that low, and just chew up CPU time pointlessly. Before we go back to TroutWars.cpp to implement the code for **CStarfield** and **CStar**, there's one last thing to add. Just below the last **#define** line, add these three:

```
inline int randrange(int lo, int hi) {
    return rand() / (RAND_MAX / (hi - lo + 1)) + lo;
}
```

That's a very simple inline function that allows us to say **randrange(1,10**) and get a number in that range returned - we'll be using that soon.

Now, back into TroutWars.cpp. There are quite a few functions we need to implement here, but first we'll do the easy things: create and destroy the star field, and move and draw the stars. In the CTWGame() function, add this line of code beneath the "new CPlayer(this)" line:

```
Starfield = new CStarfield(this);
```

Then, into the ~**CTWGame**() function, add this line after the **delete Player;** line:

```
delete Starfield;
```

Drawing the stars is done in the **DrawScene**() function. Add these two lines after the call to **ClearScreen**():

```
Starfield->Move();
Starfield->Draw();
```

That's all the periphery done, so we can now actually look at implementing the **CStarfield** and **CStar** classes. The first function is the **CStarfield** constructor, which needs to create all the stars and load the star textures. When creating each star, we need to assign it a starting co-ordinate, speed, and type, but I want to do something a little more cunning than that. As we have bright, medium, and dark stars, we're going to have the bright stars moving faster than the dark ones, with the medium stars in between.

Here's how that looks in code, with comments from me throughout:

```

CStarfield::CStarfield(CTWGame* parent) {
    // keep a local copy of the game
    game = parent;

    // sfcStars is a pointer to a pointer again, as it's an
array
    sfcStars = new SDL_Surface*[3];
    // load the three star textures
    sfcStars[STAR_DARK] = game->LoadImage
("stardark.bmp");
    sfcStars[STAR_MEDIUM] = game->LoadImage
("starmedium.bmp");
    sfcStars[STAR_BRIGHT] = game->LoadImage
("starbright.bmp");

    // load 60 dark stars
for (int i = 0; i < 60; ++i) {
  CStar* newstar = new CStar;
      // speed will be between 1 and 3
      newstar->speed = 1 + rand()%2;
      // start them between 0 (far-left) and
SCREEN_WIDTH + 500, which puts them
      // up to 500 pixels off the right edge of the screen.
      newstar->xpos = randrange(0, SCREEN_WIDTH +
500);
      newstar->ypos = randrange(0,
SCREEN_HEIGHT_TRIMMED);
      newstar->type = STAR_DARK;
      // add  to the array
```

events directly. At this point, however, we don't have any events that need sounds to go along with them, so we'll just play back our sound randomly. Enter in these three lines before the call to **SDL_GetKeyState()** in the **Play()** function of TroutWars.cpp:

```
if (!(rand() % 100)) {
    Mix_PlayChannel(-1,mixSound,0);
}
```

The first line calls the **rand()** function, which generates a random number. This is then used with the modulus (**%**) operator to calculate the remainder when that number is divided by 100. For example, 12 % 5 = 2 (*ie* 12 / 5 = 2, remainder 2), 25 % 4 = 1 (*ie* 25 / 4 = 6, remainder 1), 25 % 5 = 0 (*ie* 25 / 5 = 5, remainder 0). Using modulus is a great way to get a number between 0 and n-1, where n is the divisor. In our code, we use 100, which means **rand() % 100** will return a number between 0 and 99. As this is all in an *if* statement, it's important to remember that 0 normally equates to false and anything else equates to true. Note, however, that there's an exclamation mark before the **rand()** call – that reverses everything so that 0 equates to true and everything else equates to false. What this means is that 99 times out of 100, our call to **Mix_PlayChannel()** will not be executed, which should give us a nice irregularity to the sound playing. The **Mix_PlayChannel()** function itself is where the sound is actually played, and it takes three parameters: the channel to play on, the sound to play, and the number of times to loop. In our code we specify **-1** as the channel, which instructs SDL to use the first free channel, specify our sound as parameter two, and pass **0** as the last parameter so that it plays just once. Depending on how fast your computer is, 99 times out of 100 may mean you rarely hear the sound or it plays very regularly – increase 100 to 200 to make it twice as rare, or decrease it to make it more frequent. Don't forget to copy **laser1.wav** from your coverdisc to your working directory, then compile and run the code.

As always, I recommend that you check your code against mine on the coverdisc – I tend to fix minor bugs and/or clean up things and/or add comments as I go along; so, if you find that you have any problems with your own version, please double-check it before posting on the forums. LXF

NEXT MONTH

We have sound, we have music, we have graphics of sorts, and we even have a scrolling background. Next month we're going to be working on enemies, which also means our player will need something to shoot them with; and – most importantly – we'll be looking at how collision detection works. See you then for more *Trout Wars*!

```
     stars.push_back(newstar);
     }

   // load 70 medium stars
   for (int i = 0; i < 70; ++i) {
  CStar* newstar = new CStar;
     // speed between 1 and 5
     newstar->speed = 1 + rand()%4;
     newstar->xpos = randrange(0, SCREEN_WIDTH +
500);
     newstar->ypos = randrange(0,
SCREEN_HEIGHT_TRIMMED);
     newstar->type = STAR_MEDIUM;
     stars.push_back(newstar);
     }

   // load 30 bright stars
   for (int i = 0; i < 30; ++i) {
  CStar* newstar = new CStar;
     // speed between 4 and 12
     newstar->speed = 4 + rand()%8;
     newstar->xpos = randrange(0, SCREEN_WIDTH +
500);
     newstar->ypos = randrange(0,
SCREEN_HEIGHT_TRIMMED);
     newstar->type = STAR_BRIGHT;
     stars.push_back(newstar);
   }
}

```

Note that the three .bmp files are included on your coverdisc, and are simple little dots. It's more flexible to do it this way as opposed to drawing individual dots on the screen, as it means you can change the textures to your own version easily. The destructor for **CStarfield()** needs to pretty much do the reverse: delete all the star objects, empty the array, and free up the star textures.

It is slightly more complicated because of the need to delete the stars. The complication here is that they are stored in a vector, and you can't just delete from a vector freely as it automatically adjusts the rest of the queue and you may be stuck accessing invalid memory space. The correct way to do it is shown in the following code – you use what's called an iterator, which is essentially an index into the vector; then you move through the vector element by element and delete what the iterator points at. Finally, once all the memory is freed, you call the **clear()** function of the vector, which removes all the array elements. Here's the code:

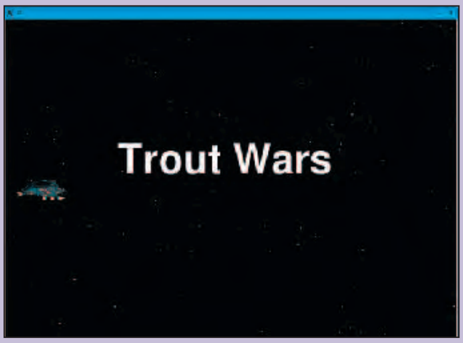
```
CStarfield::~CStarfield() {
   for (std::vector<CStar*>::iterator iter = stars.begin();
iter != stars.end(); ++iter) {
       delete (*iter);
   }

   stars.clear();

   SDL_FreeSurface(sfcStars[STAR_DARK]);
   SDL_FreeSurface(sfcStars[STAR_MEDIUM]);
   SDL_FreeSurface(sfcStars[STAR_BRIGHT]);
delete[] sfcStars;
}
```

The line **delete (*iter)** is the key – an iterator by itself is just a number, really. But when you treat it like a pointer and dereference it, what you get is the element in the array at that position, which is what we want to free.

Next up, the code to draw the star field. This needs to cycle through all the stars in the array, and call the **DrawImage()** function we created last issue for each star as the correct X and Y co-ordinates. In code, that looks like this:

```
void CStarfield::Draw() {
  for (unsigned int i = 0; i < this->stars.size(); ++i) {
     game->DrawImage(sfcStars[this->stars[i]->type],
this->stars[i]->xpos, this->stars[i]->ypos);
  }
}
```

The only thing that might briefly fox you there is the first parameter to **DrawImage()**. If you recall (or if you take a moment to look at the source code), the first parameter is the **SDL_Surface** to draw onto the main game window. For our stars, we have an **SDL_Surface** array called **sfcStars**, and each star has a **type** variable that points to the element in **sfcStars** that should be drawn – hence the first parameter. Everything else should be obvious, however.



This might look identical to a screenshot from last issue, but trust me – it's full of stars!

Finally, the code to move the stars:

```
void CStarfield::Move() {
  for (unsigned int i = 0; i < this->stars.size(); ++i) {
    this->stars[i]->xpos -= this->stars[i]->speed;
    if (this->stars[i]->xpos <= 0) this->stars[i]->xpos =
randrange(SCREEN_WIDTH, SCREEN_WIDTH + 500);
  }
}
```

The same loop is used – it goes through every star. However, this time we move the stars to the left by an equal amount to their speed: the faster a star is, the more it moves to the left each time **Move()** is called. Then, if the xpos of a star (its X co-ordinate) is less than or equal to 0, which essentially makes it invisible, its xpos gets reset to be between the far right of the screen and the far right of the screen + 500 pixels. If this was statically set to **SCREEN_WIDTH** as opposed to using the call to **randrange()**, the stars would always stay in the same position relative to other stars of their brightness – you would see three fixed panes, as it were, moving together. By randomising each star's position, there are no fixed positions, which adds to the look immensely.

That wraps up our star field code, and as you can see it's very easy to use – you create it, delete it, move it, and draw it. Internally, yes, a lot goes on, but the joys of object-oriented programming mean the outside world doesn't have to see the complexity.

PROCESS CONTROL

Practical PHP programming

On the CD and DVD

We got all forked out last month covering the first half of the POSIX and process control functions, but Paul Hudson isn't alarmed...

Outside it's dark, The streets are silent apart from the odd bird out looking for a late-night snack, but the only other people awake at this time are either enforcing the law, or on the run from it – it's 4am, and you're reaching all-new depths of desperation. If you think this sounds like the opening of a cheesy horror novel, you're about half-right – instead it's the situation where programmers make their biggest mistakes. We've all experienced it. At 4:01am, just as you're about to give up trying to solve a problem, you get a flash of insight – an all-new, previously unthought of idea that solves your problem completely and utterly.

Sadly, although these fixes might look good, they rarely are, and one of the biggest culprits in this situation is the functionality we're going to be looking at this issue. Last issue we looked at basic process control – forking child processes, and handling signals. This issue we're going to be looking at alarms, which allow you to set a timer on the system that will execute a callback function in your PHP script after the requested time has expired.

Now, here's the catch: all too often people use process alarms add a weird form of multitasking: they set their code off doing one thing, then set callback functions and alarms to perform other functionality "at the same time". This isn't how alarms work, and it usually all ends in tears. Before you go any further, understand these two things:

1 One PHP process only executes one line of code at a time, no matter how many alarms you set; and

2 The process forking code we looked at last month is the best way to do multiprocessing, no matter what anyone else tells you.

With that in mind, it's time to rock and roll...

Wake up call

Consider this situation: how would you go about making your application execute an action in three seconds' time? The obvious answer is using code like this:

```
<?php
 /// some stuff here...
 sleep(3);
 /// some more stuff here...
?>
```

However, the problem there is that the **sleep()** call *blocks* – it pauses execution of the script and doesn't allow it to do anything until the process wakes again after three seconds. This is not a problem in some rare situations, but more often people want feedback and would rather not have their applications lock-up for several seconds at a time. Instead what we really want is a *non-blocking* way to time three seconds, so that during the wait our program can be executing other things.

This is where the **pcntl_alarm()** function comes in. Each process is allowed to have one alarm timer in place, and you place the timer by calling **pcntl_alarm()** and passing the number of seconds you want the alarm to wait. When the time is up, your application will be sent the **SIGALRM** signal, which you can then handle.

Try this code out - it builds upon the code presented last issue so that it handles **SIGALRM** along with the other signals:

```
<?php
 declare(ticks=1);

 function signal_handler($signal) {
  switch($signal) {
   case SIGTERM:
    echo "Caught SIGTERM\n";
    exit;

   case SIGQUIT:
    echo "Caught SIGQUIT\n";
    exit;

   case SIGINT:
        echo "Caught SIGINT\n";
        exit;

      case SIGALRM:
        echo "Caught SIGALRM!\n";
        break;
    }
  }

  pcntl_signal(SIGTERM, "signal_handler");
  pcntl_signal(SIGQUIT, "signal_handler");
  pcntl_signal(SIGINT, "signal_handler");
  pcntl_signal(SIGALRM, "signal_handler");

  pcntl_alarm(3);

  while (1) {

  }
?>
```

When you run that code, it should do nothing for three seconds, then print the message, "Caught SIGALRM!," then do nothing until you press **Ctrl-C**. This shows two things: setting up an alarm is a cinch, and also that once your alarm 'rings,' it is not reset – you need to call **pcntl_alarm()** again to make it go off again. Thus, the **SIGALRM** case in the previous script can be amended to this:

```
case SIGALRM:
  echo "Caught SIGALRM!\n";
  pcntl_alarm(3);
  break;
```

This time, when the alarm rings it sets another alarm, effectively making the alarm go off every three seconds until the script is killed. Now, the important thing about all this is that while the alarm is waiting to go off in the background, PHP continues to whiz through the infinite while loop. Thus if you had other code in there, such as handling a GUI, it would carry on being executed while the alarm is in place.

Here's where the confusion arises: when the alarm is hit, it calls your signal handling function. However, at that point PHP stops whatever it was doing in the while loop, and hands control over to the callback function. It then executes all parts of the callback function (the echo, the resetting of the alarm, and the break) before returning to the while loop. At no time are processes spawned – there's no multiprocessing going on at all. At best it's just 'virtual multitasking,' in that it appears to do two things at once.

Positively POSIX

One of the nice things about working on a Unix is that you know it has a set number of commands that are pretty much guaranteed to work as you'd expect them to. In fact, deep down this is one of my favourite aspects of Unix – I can *SSH* (or, gulp, *Telnet*) into an unknown system, run a few commands, and have a pretty solid grasp of where I am and what I have available. This is no happy co-incidence, and is primarily down to the POSIX specification and others like it. POSIX itself stands for the Portable Operating Systems Interface (POSI), though at the time, Unix had such influence in the world that no one was surprised when an X was added to the end.

PHP has quite a few functions that come under the POSIX header, of which all are thin-glue wrappers around their C equivalents. This effectively turns makes quite a few standard C functions available in scripts, which is what we'll be looking at here. Although I'll only be covering a handful of the 32 POSIX functions, they are the ones that are most immediately helpful. In order, these are:

- **posix_getlogin():** Get the login name of the user that started the current process. This takes no parameters and returns a username.
- **posix_getpid():** Get the process ID of the current process. Takes no parameters and returns the PID as an integer.
- **posix_getpwnam():** Get detailed information on a username. Takes a username as a parameter, and returns an array full of user information.
- **posix_kill():** Send a signal to another process. Takes a PID and a signal as its parameters, and returns true on success.
- **posix_times()**: Return CPU usage for this process. Takes no parameters and returns an array of information.
- **posix_uname():** Get system information. Takes no parameters and returns an array of information.

Three of those return arrays full of information, and the easiest way to see what it contains is simply to call them and use **var_dump()** to see what was returned, like this:

»

IN THE KNOW

Describing the pcntl_exec() function

Last issue, I reviewed the book *Core PHP Programming, 3rd Edition*, giving it the fairly low score of **4/10**. Amongst the problems cited with it were that the description of the **pcntl_exec()** function was incorrect, and one keen-eyed reader has written to me asking that, if the description was wrong, how *should* the function be used? Although I wasn't planning on covering it, here's a brief description.

In the same way that the **passthru()** function executes an external program, the **pcntl_exec()** program also executes an external program. However, it executes it using the current address space, which means that the PHP script *stops* executing and is replaced by the external program. This script should make it clear:

```
<?php
  echo "Checkpoint 1\n";
  passthru('/usr/bin/uptime');
  echo "Checkpoint 2\n";
  pcntl_exec('/usr/bin/uptime');
  echo "Checkpoint 3\n";
?>
```

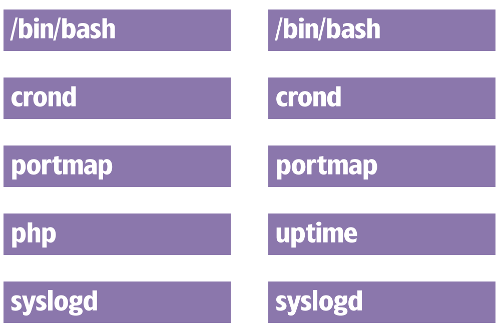
Running that script will output something like the following:

```
Checkpoint 1
 16:34:42 up 2 days,  5:06,  2 users,  load
average: 0.02, 0.10, 0.06
Checkpoint 2
 16:34:42 up 2 days,  5:06,  2 users,  load
average: 0.02, 0.10, 0.06
```

So, **Checkpoint 1** is reached, **passthru()** is called so /usr/bin/uptime is called and its output is printed out (just calling **exec()** wouldn't automatically print the output); **Checkpoint 2** is reached, **pcntl_exec()** is called and so calls /usr/bin/uptime again, then... nothing? Note that **Checkpoint 3** is never reached, because the call to **pcntl_exec()** effectively terminates the PHP script.

Hopefully that should make the difference quite clear – **pcntl_exec()** is quite a special function, and, like most special functions, you "just know" when you need it.

PROCESS



On the left you can see some of the running processes on a machine before a call to pcntl_exec(). On the left, you can see the same processes after the call – note that php is now missing, as it has been replaced by uptime.

«

```
<?php
  var_dump(posix_getpwnam("paul"));
  var_dump(posix_times());
  var_dump(posix_uname());
?>
```

Here's what that outputs on my screen, with much of the whitespace removed to save space:

```
array(7) {["name"]=>string(4) "paul" ["passwd"]=> string(1)
"x" ["uid"]=> int(501) ["gid"]=> int(501) ["gecos"]=> string(11)
 "Paul Hudson" ["dir"]=> string(10) "/home/paul" ["shell"]=>
string(9) "/bin/bash" }

array(5) { ["ticks"]=> int(448754096) ["utime"]=> int(0)
["stime"]=> int(0) ["cutime"]=> int(0) ["cstime"]=> int(0) }

array(5) { ["sysname"]=> string(5) "Linux" ["nodename"]=>
string(7) "shazbat" ["release"]=> string(10) "2.6.3-4mdk"
["version"]=> string(30) "#1 Tue Mar 2 07:26:13 CET 2004"
["machine"]=> string(4) "i686" }
```

About half of those fields are self-explanatory, but many aren't. Here's what they all mean:

- **Name** – username
- **Passwd** – password; note the **"x"** because I'm using shadow passwords
- **UID** – User ID
- **GID** – Group ID
- **GECOS** – finger information; my full name
- **Dir** – my home directory
- **Shell** – my default shell program - bourne again, baby!
- **ticks** – number of clock ticks since last reboot
- **utime** and **stime** – user time and system time used by the current process
- **cutime** and **cstime** – user time and system time for this and any child processes
- **sysname** – OS name; equivalent to **uname -s**
- **nodename** – hostname; equivalent to **uname -n**
- **release** – kernel version; equivalent to **uname -r**
- **version** – kernel version; usually this is the time it was built; equivalent to **uname -v**
- **machine** – architecture, *eg* i686; equivalent to **uname -m**

As you can see, there's a whole lot of information contained in those three innocent looking functions, and it's easily argued that these functions give away information best kept away from prying eyes, particularly given that no controls are in place even when safe mode is activated. If it's too rich for your blood, configure PHP with **--disable-posix** – yes, *"baby"* and *"bath water"* come to mind, but so do *"better safe"* and *"than sorry"*.

Now, some real code to give the other functions an airing:

```
<?php
  $login = posix_getlogin();
  $mypid = posix_getpid();

  $rand = rand(1,2);
  if ($rand == 2) {
    echo "Process $mypid owned by $login has been selected
for death!\n";
    posix_kill($mypid, SIGKILL);
  } else {
    echo "Process $mypid owned by $login lives on!\n";
  }
?>
```

That will kill the current process 50 per cent of the time, printing out the PID and the name of the login account that started it. Note that you may get something like **"Process 24067 owned by lives on!"** – that is, there will be a blank where the username should be. This is usually caused by running the script from within an X terminal. The problem here is that login information usually resides in the file *utmp* (it's binary, so don't try reading it by hand – use the program *utmpdump*), which probably lives in the /var/run/ directory. If a terminal doesn't register itself with this file, the login information won't be available, hence the missing information in the output. Running the same script from a standard, non-X terminal should be fine.

Error handling

As the PHP POSIX functions are essentially rebadged C functions, they aren't quite so easy to debug as 'normal' PHP functions. However, there are two functions to help you out: **posix_get_last_error()** and **posix_strerror()**. The first returns an error number, or **0** if nothing was wrong. You can then pass this error number into **posix_strerror()**, which returns the textual representation of the error (or **"Success"** if the error number was **0**). So, if you want to check for an error when calling **posix_kill()**, you'd use code like this:

```
<?php
  posix_kill($mypid, SIGKILL);
  $errno = posix_get_last_error();
  if ($errno) {
    echo "Error encountered: ", posix_strerror($errno), "\n";
  } else {
    echo "No errors encountered.\n";
  }
?>
```

Of course, if the call to **posix_kill()** is successful, **"No errors encountered"** won't be printed out, as the process will be killed.

Be a control freak

To finish our excursion into the POSIX and process control functions, we're going to look at how to take complete control over another process. What we're going to be doing here is open a process and set up a read pipe and a write pipe so that we can communicate bi-directionally with it, so be prepared for a little thinking!

The functions we'll be using to open and close the process handle are **proc_open()** and **proc_close()**, the first of which is a little complicated to use. Here's the function prototype straight from the PHP manual:

```
resource proc_open ( string cmd, array descriptorspec, array pipes)
```

It returns a resource, which is a running process. You need to pass this into **proc_close()** at the end of the script, but that's about all. The second parameter is where you set up the communication layer between the parent process (our script) and the child (the process we're going to launch). This needs to be a two-dimensional array, with at least one entry in order to provide communication. Each value in this array is itself an array with two elements: the first is a string that needs to be either **pipe** (communicate through a pipe) or **file** (communicate through a file), and the second needs to be either **"r"** for 'read', **"w"** for 'write', or a filename.

Now, here comes the magic. If you've ever done advanced console work, you'll have seen commands like this:

```
grep * 2>&1
```

That would redirect the errors to standard output (**stdout**). The reason for this is because standard input (**stdin**) is given the number **0**, **stdout** is **1**, and standard error (**stderr**) is **2**. These same numbers are needed in our array – we've already looked at the values, but the *keys* need to be these numbers, as they relate to the child process. That is, **0** means **stdin**, so that's the **pipe** the child process will read from. So, to set up a write-only **pipe** (a **pipe** that the parent can only write to, meaning that the child would read from), we'd use this:

```
$descriptorspec = array(
  0 => array(“pipe”, “r”)
);
```

Similarly, to set up a **pipe** for the child to write to (parent read-only) and a file where the child should save its errors, we'd use this:

```
$descriptorspec = array(
  0 => array(“pipe”, “r”),
  2 => array(“file”, “/tmp/myerror.log”, “a”)
);
```

Note the extra "mode" parameter – **"a"** – required to set how the file should be worked with. That covers parameters one and two of **proc_open()**. The last parameter is an array where the created **pipes** can be stored, so just pass in a fresh variable.

Now, here's some code so you can see it all in action:

```
<?php
  $descriptors = array(
    0 => array(“pipe”, “r”)
  );

  $process = proc_open(“php”, $descriptors, $pipes);

  if (is_resource($process)) {
    fwrite($pipes[0], “<?php\n”);
    fwrite($pipes[0], “  \$rand = rand(1,2);\n”);
    fwrite($pipes[0], “  if (\$rand == 1) {\n”);
    fwrite($pipes[0], “    echo \”Hello, World!\n\”;\n”);
    fwrite($pipes[0], “  } else {“);
    fwrite($pipes[0], “    echo \”Goodbye, World!\n\”;\n”);
    fwrite($pipes[0], “  }”);
    fclose($pipes[0]);

    $return_value = proc_close($process);
            }
?>
```

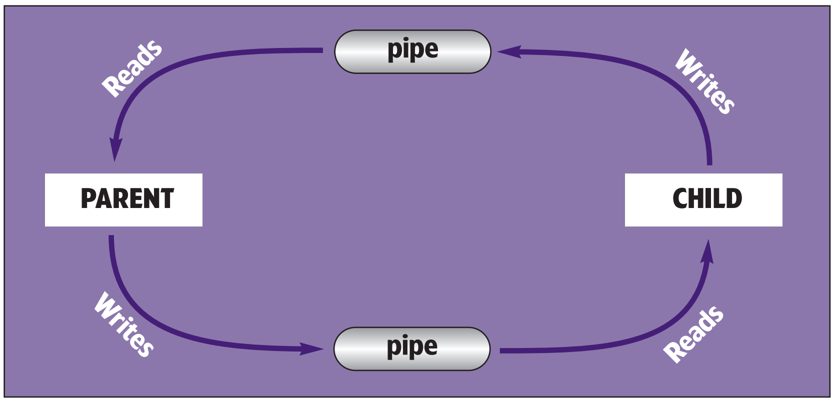
After the call to **proc_open()**, we check whether the return value is a resource or not – if not, clearly the call failed so we should do nothing else. However, if the call succeeded, we can go ahead and work with the process. The next seven lines are calls to **fwrite()**, as data is being sent to the PHP process. Note that we write to the first element of our **$pipes** array – again, remember that this was marked as **"r"** for the child, which means it's writeable for the script. Once all the writing is done, **fclose()** needs to be called on the pipe.

Finally, **proc_close()** is called to close the process. You should always **fclose()** all your **pipes** before you call **proc_close()**, otherwise you're likely to lock up the script!

Reading data back

So far we've only done writing, but it's easy to make our script move to bi-directional communication. In order to read data back, we need a second **pipe**, and also need to call **fread()** once we've finished writing. In order to get the second **pipe**, change the **$descriptors** array to this:

```
$descriptors = array(
  0 => array(“pipe”, “r”),
  1 => array(“pipe”, “w”)
);
```



The parent and child share pipes for reading and writing. The key thing to remember is that the operation is flipped, as shown – what the parent reads from, the child writes to, and vice versa.

Now, before we go any further, try running the script again. You should find that the text is no longer printed out – this is because the output of the script (where it writes to) is being piped back to our script, and we're not doing anything with it yet. So, after the call to **fclose()**, add this code:

```
while (!feof($pipes[1])) {
  echo fgets($pipes[1]);
}

fclose($pipes[1]);
```

Once we've written to the file, we now read back everything available – this loop keeps going around as long as we've not hit the end of the pipe (thanks to **feof()**). Each time there's more to read, **fgets()** is called to read in that line and echo it out, essentially printing out everything that's been sent back. Once the loop finishes, the pipe is closed, leaving the way clear for the **proc_close()** call.

Give that a try – all being well, it should work in precisely the same way as before, except now we capture input before printing it out. This means we can post-process the text, like this:

```
$output = “”;
while (!feof($pipes[1])) {
  $output .= fgets($pipes[1]);
}

$output = strtoupper($output);
echo $output;
fclose($pipes[1]);
```

Done and dusted

That wraps up our coverage of the process control and POSIX functions – we've looked at handling signals, forking children, setting alarms, running standard Unix system calls, and comprehensive process control. Hopefully you can see how useful these functions are, although their usefulness is of course limited to the CLI SAPI – I'd recommend you avoid them as much as possible for use with *Apache*, and also remember to be wary of safe mode issues.

Next issue, we're going to be moving on to greener pastures and looking at how the new *MySQL Improved* (*mysqli*) extension works. This is a complicated beastie, as it's developed independently of the core *MySQL* extension we've been using so far. From *MySQL 4.1* onwards an all-new system is being used, hence the *MySQLi* extension – I suggest you download and install the latest 4.1 snapshot (must be greater than 4.1.1!) from the *MySQL* website and get it working before next month. LXF

NEXT MONTH

PHP has been forked! Hardened PHP makes the programming language secure to its core, which prompts the question: how can you write secure PHP in the first place? Find out next month!

GIMP PROGRAMMING

Writing GIMP Plugins in Perl

PART 2 You can't get far with *GIMP Perl* without knowing how to access *The GIMP*'s internal functions, not to mention letting your user configure your script, says Michael J Hammel.

TIP

***GIMP1.2* and *GIMP 2.0* both provide four native programming interfaces: C, Perl, Python and *Script-FU* (a subset of *Scheme*). The design of *The GIMP* allows further language extensions to be added fairly easily. This series on *GIMP Perl* will not be the only language covered in this column. You can expect to see more on both the Perl and C interfaces in the coming months.**

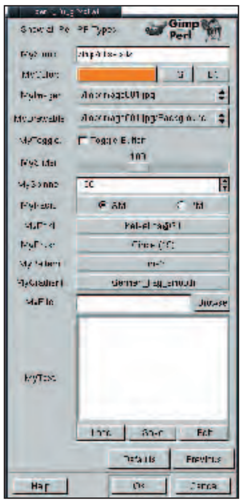


Fig1 **A *GIMP Perl* dialog with all possible PF_TYPEs used in the parameter list of the register() function call.**

When you cast your mind back to last month's tutorial, you'll recall that we introduced you to *GIMP Perl,* one of the four programming interfaces available from *The GIMP* for creating plugins. Nearly all *GIMP* plugins provide a user interface where users can set options before processing. *GIMP Perl* provides a simplified *GTK+* interface through the *GIMP::Fu* Perl module. This module offers a variety of input options, all of which are accessible through the register() function which we introduced briefly in the previous article in this series.

Aside from the *GIMP::Fu* module, *GIMP Perl* also provides both procedural and object-oriented interfaces into the *GIMP* function library. This library – known more commonly as the *Procedural Database* or *PDB* – is key to understanding how to making the most of *GIMP Perl.*

In this months tutorial, we'll take a detailed look at both *GIMP::Fu* and the *PDB* while using the second of our two example scripts – GFXLayerSave.pl - to demonstrate their use.

Building a better interface

GIMP::Fu is a wrapper around *The GIMP*'s C API that also provides access to the *GTK+* toolkit – the software which provides buttons, menus and text input fields for *GIMP* and the GNOME desktop. In a way, this is the closing a small circle of life for this software: *GTK+* began life as a tool specifically built for *The GIMP*, became a software entity of its own used by many applications and even GNOME, and now provides script language bindings that can be used to create plugins for its original parent – *The GIMP*.

In last months article, we discussed the use of the 9th argument to the **register()** call in a *GIMP Perl* script. This argument is a parameter array with each element being another array of values. The values define an input option for a user interface feature such as a text entry field, a color selection dialog or a set of radio buttons. The values also provide default settings, valid ranges and various other options.

The first part of the register function is laid out like this:

```

register(
   "function_name",
   "blurb", "help",
   "author", "copyright",
   "date",
   "menu path",
   "image types",
   [
     [PF_TYPE, name, desc, default, extra_args],
     [PF_TYPE, name, desc, default, extra_args],
     ...
   ],
   ....

```

GIMP SCRIPTS

On the *LXF* website at http://www.linuxformat.co.uk/gimp/55.zip, you will find the following files that are mentioned in the course of this tutorial. The two main files were included on last month's discs.

pf_all.pl
GFXLayerSave.pl.commented
GFXLayerSave.pl
GFXOffsets.pl.commented
GFXOffsets.pl

We will include these files on next month's discs as well, for those who do not have access to an Internet connection.

Note that argument nine of this function is enclosed in brackets, signifying an array. This example shows two elements in the parameter array. Each element specifies a parameter type (PF_TYPE in the example) followed by a name and description (both are required). After the description come optional default settings and any additional arguments specific to the parameter type.

The following list shows the set of possible parameter types and what kind of user interface option they provide. **Fig1** below left shows a *GIMP Perl* dialog window with all possible parameter types (except **PF_CUSTOM**) displayed.

```
File: scripts/pf_all.pl
```

shows the code that was used to generate **Fig1**.

As you can see from his simple **pf_all.pl** script, *GIMP:Fu* offers a wide variety user interface options: **PF_INT8, PF_INT16, PF_INT32, PF_INT, PF_FLOAT, PF_STRING, PF_VALUE**

These all provide text input fields, since Perl doesn't differentiate between strings and numeric values. Offering different types that map to a single user interface type allows other plugins to use the value in a language-appropriate way:

PF_COLOR

This provide users with the option of using a color selection dialog to choose a colour.

PF_IMAGE and PF_DRAWABLE

Provide menus of the currently open images and drawables (masks, channels or layer).

PF_TOGGLE, PF_BOOL

These provide a single button that will return either TRUE (if selected) or FALSE (if not selected). The default value for this can be TRUE, FALSE, 1, or 0. The description is also used for the toggle-button label.

PF_SLIDER

Displays a horizontal scale. To set the range and step size append an array in the form

```
[range_min, range_max, step_size, page_increment, page_size]
```

as an extra argument to the parameter array. Default values will be substituted for missing entries, for example:

```
[PF_SLIDER, "alpha value", "the alpha value", 100, [0, 255, 1] ]
```

which sets the min, max and step size and uses *GTK+* defaults for page increment and page size in the slider.

PF_SPINNER

Provides a spinner widget. Ranges are specified in the same manner as the **PF_SLIDER** parameter type.

PF_RADIO

The extra argument field must refer to an array filled with Option-Name = Option-Value> pairs. *Gimp::Fu* translates this to a series of buttons, laid our horizontally, one for each pair. For example:

```
[PF_RADIO, "Hour", "AM or PM", 1, [AM => 1,  PM => 2]]]
```

draws two buttons: **AM** and **PM**. If **AM** is selected, the callback will receive a value of **1**. If PM is selected, it will receive a value of **2**.

PF_FONT

Provides a font selection option. This feature returns a X Logical Font Descriptor (XLFD) to the callback. The default argument, if specified, must also be a full XLFD specification or a warning will be printed.

PF_BRUSH, PF_PATTERN, PF_GRADIENT

These provide access to brush, pattern and gradient selection dialogs. The value returned can be used in the appropriate tool selection function.

PF_CUSTOM

For scripts requiring a non-standard-widget. See the pod documentation for details on how to use this.

PF_FILE

Provides a text field and button for browsing the file system. Its use is primarily for selecting files but is not limited to this.

PF_TEXT

Like PF_STRING, but presents a multi-line field for text entry along with buttons for saving the text, loading text from a file and editing the text with the user defined default text editor.

The name field for a parameter entry is used as the label displayed next to the widget (widgets are buttons, menus, and essentially any other part of a window). The description field is used in the *PDB*. A special case is the PF_TOGGLE type, which uses the description field as the text for the toggle button.

The order of the parameter elements defines the order they will be passed to your callback function. Remember that your callback function is the one that does the real work and is referenced as the last argument in the call to the **register()** function.

Some parameter element types include option arguments. These option arguments (which are not actually optional if the parameter type requires them) can be single values or arrays. For example, the PF_RADIO type requires an associative array in the option arguments field, with each element of that array describing the label and value associated with a single radio button. The PF_SLIDER parameter type also takes an array of values defining range settings.

The simplicity of *GIMP::Fu* is that the entire user interface is defined as a series of arrays inside a single array, all wrapped inside a single function call. This simplicity comes at a price, however. There is little flexibility in the layout of the user interface – all components are aligned vertically in the plugin dialog. There is little interactivity with the dialog as well. Callbacks for widgets are not configurable, so interactivity is restricted to what *GIMP Perl* has built in – you can't, for example, allow a user to draw shapes in a preview in a *GIMP Perl* plugin, while you can do this with the C API.

Despite such restrictions, *GIMP Perl* still provides adequate functionality for most scripts. Scripts are – after all – primarily quick methods of repeating a series of steps that users find they do often. *Adobe Photoshop* users might consider this similar to the use of Actions, but scripting offers more flexibility than Actions, while not being quite as flexible as compiled plugins.

Who to call and how to call 'em

With the dialog design firmly in mind, we can now look to how to talk to *The GIMP* itself. There is a core set of features provided by *GIMP* for plugins. This core set includes things like retrieving information about layers, accessing tools from the Toolbox and dealing with cut and paste. Beyond this core set, any plugin or script that calls **register()** provides a function that will be added to the *PDB*.

The GIMP's plugin API is the *Procedural Database*, or *PDB*. The *PDB* gives plugins written in any language access to both internal functions of *The GIMP* as well as features provided by other plugins. Any plugin that has registered its callback function can have that function called by any other plugin.

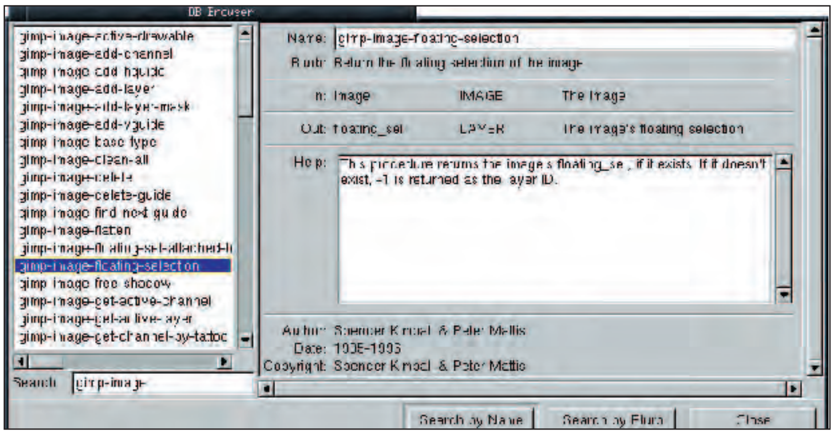
To find a registered function name, you need you start with the *DB Browser*. This is a dialog found via the Xtns>DB Browser menu option in *The GIMP*'s Toolbox. This dialog offers a searchable, scrolled list of functions on the left, and information about the currently selected function on the right.

At the bottom left of this dialog is a text input field. Typing a word – or even a few letters – here and hitting **Enter** will limit the set of functions displayed in the scrolled list to any that contain that string of characters. If you want to see the whole list, clear the text field and hit **Enter**.

Clicking on a function in the list will bring up information about that function on the right side of the dialog. Functions can have input arguments – values a plugin will pass to it – and output arguments that are returned to the calling program. They are not required to have either, however. It possible to write a function that runs the entire list of layers of all open images, for »

TIP

The *GIMP Perl* function name provided as the first argument to the call to register() is the name under which your plugin will be registered in the *GIMP* database. If this name doesn't start with 'perl_fu_', 'file_', 'plug_in_' or 'extension_' then the 'perl_fu_' prefix will be added. If you don't want this, prefix your function name with a single '+'.



DB Browser

gimp-image-active-drawable
gimp-image-add-channel
gimp-image-add-hguide
gimp-image-add-layer
gimp-image-add-layer-mask
gimp-image-add-vguide
gimp-image-base-type
gimp-image-clean-all
gimp-image-delete
gimp-image-delete-guide
gimp-image-find-next-guide
gimp-image-flatten
gimp-image-floating-sel-attached-t
gimp-image-floating-selection
gimp-image-free-shadow
gimp-image-get-active-channel
gimp-image-get-active-layer
gimp-image-get-channel-by-tattoo

Search: gimp-image

Name: gimp-image-floating-selection
Blurb: Return the floating selection of the image
In: image IMAGE The image
Out: floating_sel LAYER The image's floating selection
Help: This procedure returns the image's floating_sel, if it exists. If it doesn't exist, -1 is returned as the layer ID.
Author: Spencer Kimball & Peter Mattis
Date: 1995-1996
Copyright: Spencer Kimball & Peter Mattis

Fig2 **The *DB Browser* shows what functions are available to plugins. Note the input and output arguments, and the help text that describes these arguments.**

TIP

There is an additional browser for *PDB* that comes with *GIMP Perl* called the *PDB Explorer*. This dialog is essentially the same as the *DB Browser* but is Perl based. It is little more informative than the the *DB Browser* though it does show function names with underscores instead of dashes.

« example. Such a function would not need to be passed any arguments as the set of images and set of layers for each image are available from *The GIMP*'s core set of functions.

The *DB Browser* lists function names the way they are called using the procedural interface. The only difference is that the *PDB* lists function names with dashes between words in the name while *GIMP Perl* requires you to use underscores instead. The procedural method requires all arguments listed in the *DB Browser* to be passed in the function call while the object-oriented method allows you to leave out arguments that are handled by the object. For example, the procedural call to retrieve the **id** of the current floating selection would be

```
$float_id = gimp_image_floating_selection($image_id);
```

while the object-oriented method would attach this call to an image object, allowing you to leave off the **gimp_image_** prefix:

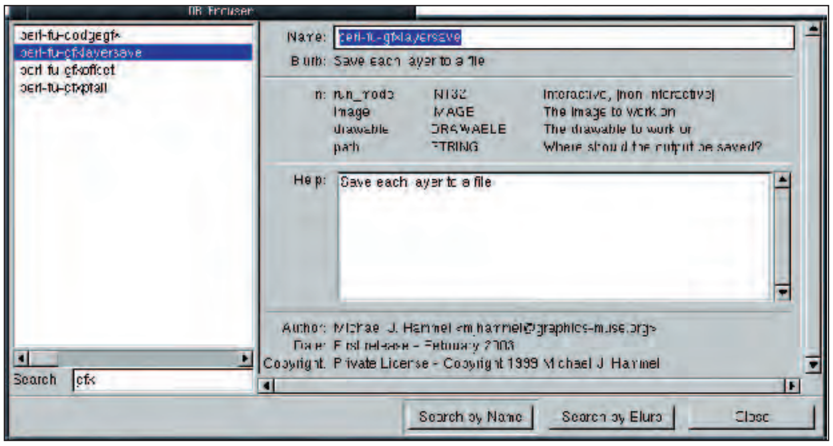
```
$float_id = $image_id->floating_selection();
```

where **$image_id** is the current image **id** and was passed to your callback function as the first argument (and the current drawable as the second argument) as long as you specified your plugins menu location to be headed by **<Image>**. *GIMP Perl* is smart enough to know that if the method requested doesn't exist for that object it can call the procedural version of that function instead. Note that if you specify your plugin to fall under **<Xtns>** instead, you will not be passed any image or drawable arguments. Remember that the menu location for your plugin is a parameter in your call to **register()**, as we discussed last month.

The real world, part II

With UI and API in hand, we can return to a real-world example with more confidence. Remember last month we said there were two scripts in this project: GFXOffsets.pl and GFXLayerSave.pl. GFXOffsets was the more simple of the two, and we breezed through it fairly quickly. Now, it's time to look at the slightly more complex GFXLayerSave.pl.

```
register (
   "gfxlayersave",
   "Save each layer to a file",
   "Save each layer to a file",
   "Michael J. Hammel <mjhammel\@graphics-muse.org>",
   "Private License - Copyright 2004 Michael J. Hammel",
   "First release - February 2004",
   "<Image>/Filters/GFXMuse/GFXLayerSave",
   "*",
   [
      [ PF_STRING, "path", "Where should the output be saved?",
"/tmp" ]
   ],
   \&LayerSaveGFX_Run
);
```



```
Name: perl-fu-gfxlayersave
Blurb: Save each layer to a file

In:  run_mode    INT32      Interactive, [non-interactive]
     image       IMAGE      The image to work on
     drawable    DRAWABLE   The drawable to work on
     path        STRING     Where should the output be saved?

Help: Save each layer to a file

Author: Michael J. Hammel <mjhammel@graphics-muse.org>
Date: First release - February 2003
Copyright: Private License - Copyright 1999 Michael J. Hammel

Search: gfx
```

GFXLayerSave.pl registered its function as gfxlayersave, shown here as it is displayed in the *DB Browser*, with its translated name and input arguments.

The register function again places our plugin in the Image section under the Filters/GFXMuse menu. Remember that plugins placed under the Image menus get the active image **id** and active drawable **id** passed as the first argument to the callback routine. Our callback routine is called **LayerSaveGFX_Run** and we've requested a single UI element – a text input field that we will use as the directory to store the layers as separate images.

We've also given our plugin the name **"gfxlayersave"**, which will be translated to **perl-fu-gfxlayersave** in the *PDB*. Other plugins could call our plugin with this name, passing in the image, drawable and path values. If called this way, our single callback, **LayerSaveGFX_Run**, would be be run with those arguments *sans* the UI.

The entrance into the callback routine assigns the input values, in the order they are passed, to the variables **$img, $drawable,** and **$path**. We need the image value for this script – our goal is to find each layer in that image and save it as a separate image file.

```
sub LayerSaveGFX_Run {

   # Grab the input parameters.
   my ($img, $drawable, $path) = @_;
```

Next we do some error-checking on the supplied pathname. If the directory does not exist, we use *The GIMP*'s internal message dialog to tell the user that they made a mistake, then let the plugin exit gracefully. The user will be able to see the message even if our plugins dialog has closed.

```
   if ( ! -d $path )
   {
      gimp_message("$path does not exist.\nCreate it and try
again.");
      return;
            }
```

Note that we can search for **message** in the *DB Browser* to find out how to use this function. It takes a single text string as its only argument. And because this function is not associated with any specific *GIMP* image, it doesn't have an image or drawable input argument (see the *DB Browser* entry for **gimp_message**).

The image **id** passed in to the callback can now be used as on object identifier. We first use it to retrieve the type of image – RGB, INDEXED, and so forth. This is used to choose an appropriate filename extension later on in the script. The extension is pulled from an array we defined at the top of the script, but which we include in the code shown next below for clarity. The base type is returned as an integer so the array of filename extensions is an ordinary array. If **base_type()** returned a string, we could have used an associative array instead.

```
   @EXTS   = ( "png", "png", "gif" );
   ...
   my $imgtype = $img->base_type();
   $ext = $EXTS[$imgtype];
   my @layers = $img->get_layers();
   my $count = scalar(@layers);
```

After the filename extension is set, we use the image object again to retrieve an array of layers in the image. We'll be running through that array in a moment. We also save a count of the number of layers retrieved.

```
  Gimp->progress_init("GFXLayerSave is working...");
  my $progress_increment = 1 / $count;
  my $progress = 0.0;
```

Before we begin our loop, we let the user know something is about to happen. The call to **Gimp->progress_init()** uses the global object **Gimp**. It is similar to the procedural version of this call except we don't have to specify the **gdisplay** argument (seed the *PDB* entry for **gimp-progress-init**) since we just update the active image window. A couple of progress-related variables are set up, and we're ready to enter our loop.

```
  foreach (@layers)
  {
    $height    = $_->drawable_height();
    $width     = $_->drawable_width();
    $layername = $_->layer_get_name();
    $hasalpha  = $_->drawable_has_alpha();
    $_->edit_copy();
```

The loop iterates over all the layers in the array filled by the call to **$img->get_layers()**. Each layer object is assigned to the special Perl variable **$_**, and we use that to retrieve the height, width, and layer name as well as check to see if the layer has an alpha channel (*ie* transparency). Finally, we make a copy of the layer in *The GIMP*'s primary copy buffer.

Have you noticed that the object calls use the function names from the *PDB* but without the usual **gimp_** or **image_** prefix? This is just a shortcut that the object-based *GIMP Perl* interface provides. If you use the procedural interface instead, you will need to use the function names just as they are in the *PDB*, with dashes changed to underscores. Our next few calls use the procedural interface – we can mix both methods within a single *GIMP Perl* script if we choose.

```
    $newimage = gimp_image_new($width, $height, 0);
    $layer = gimp_layer_new($newimage,$width,$height,0,
 $layername,100,0);
    if ( $hasalpha ) { gimp_layer_add_alpha($layer); }
  gimp_image_add_layer($newimage,$layer,-1);
  gimp_image_set_active_layer($newimage,$layer);
    gimp_edit_clear(gimp_image_active_drawable
 ($newimage));
```

The above shows that we created a new image with the same height and width as our layer and that is of type **RGB** (the **0** value). We added a layer to this image (**gimp_image_new()** doesn't add any layers!) that is fully opaque. If the layer from the original image had transparency, we add it to our just created layer. We then add the layer to the new image, make it the active layer and clear it. Clearing it is necessary, because a layer is a block of memory, and unless you specifically request it, (with **gimp_edit_clear**) that block of memory might have garbage image data in it.

```
    $floatsel = $layer->edit_paste(1);
  gimp_floating_sel_to_layer($floatsel);
    $newimage->merge_visible_layers(0);
```

Remember the copy we made of the original layer a while back? We now paste it using the new layer as the destination object. This will create a floating selection, which we first need to make into a new layer, and then merge with the first layer we created in the new image.

```
    $filename = $layername;
    $filename =~ s/\s/_/g;
    $filename =~ s/\//-/g;
    $filename =~ s/\"//g;
    $filename =~ s/\'//g;
    $filename =~ s/\,/_/g;
    Gimp::Fu::save_image($newimage, "$path/$filename.$ext");
```

With an original image's current layer copied into a layer in a new image, we're now ready to save it to a file. We clean up the filename a bit, which was based on the layername. The layername can have spaces and other cruft that we don't want, so all those **=~** that pervade the above code are just a way of changing the cruft to something more filename-friendly. Then we use *GIMP::Fu* to save the image. This is a convenience function in *GIMP::Fu* that makes it very easy to save images based on their filename extension. Remember: we determined the filename extension earlier – if the extension is **.png**, we save the file and **PNG**, and so forth.

```
    $progress += $progress_increment;
    Gimp->progress_update ($progress);
    $newimage->delete();
    gimp_displays_flush();
  }
  gimp_message("GFXLayerSave completed successfully.");
```

The loop ends with an update to the progress bar in our current image window (image windows are also known as Canvas windows to avoid confusion with the often overused term "image"). After that update we delete the newly created image window which was used as a temporary holder for the copy of our current layer. Then the *GIMP* windows are 'flushed', which causes them to be updated to reflect any changes (there should only be the progress update in this case).

After the loop completes, a final message window is displayed letting the user know the layers have been saved to individual files. All plugins should return **0** (zero) when they are complete so that *GIMP Perl* will know to close the window and let *The GIMP* clean up after itself internally.

This particular script has only the most simplistic user interface. It makes no changes to any Canvas windows either. So from a user perspective, it looks like it actually does very little. This is why the progress updates and calls to **gimp_message()** are so important. Other scripts will produce visible changes, so the use of **gimp_message()** may not be as important.

One important note to remember for plugins that do make changes to images: be sure to include a call to **gimp_undo_push_group_start()** at the start of your callback function and a call to **gimp_undo_push_group_end()** at the end of this function. Doing so will allow the user to use **Ctrl-Z** just once to undo whatever your plugin does, even if multiple changes are made by the plugin. Since no changes were made to any layers in this script, we didn't waste any undo levels with calls to these two functions.

There isn't much to this script and it would be easy – especially after reviewing the myriad of functions available in the *PDB* – to extend this script to do much more clever things. But this is a good start, and an easy way to see how to make use of *GIMP::Fu* and the *PDB* (via the *DB Browser*) to get the most out of scripting in *The GIMP*. LXF

FURTHER INFORMATION

To find more information on *GIMP Perl*, try Marc Lehman's original documentation at www.goof.com/pcg/marc/gimp.html. Information on *GIMP Perl* for *GIMP 2.0* is being laid out by Seth Burgess at www.gimp.org/~sjburges/perl/gimp-perl-faq.html

TIP

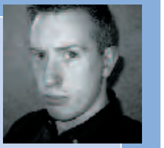
A word about modifying your script on the fly: You can update your script while *The GIMP* is running as long as its calling parameters (such as the user interface) don't change. Just make your updates to the callback function and drop the script back in $HOME/.gimp-1.2/plugins (for GIMP 1.2, GIMP 2.0 uses .gimp-2.0 instead). You can then select it from the menus immediately to test your changes. However, if the script has not been registered yet you must restart GIMP before you can use it.

Answers

If you are really stuck and the HOWTOs yield no good result, why not write in? Our resident experts will answer even your most complicated problems!

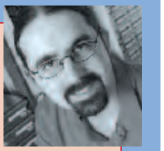
Our experts

Whatever your question is, we can find an expert to answer it – from installation and modem woes, through difficult distros to network administrations, we can find the answer for you – just fire off a letter or email and it'll all be taken care of.



LXF answers guy **David Coulson** is a networking and security guru with plenty of sysadmin experience to boot.

NIck Veitch is the editor of the magazine, and answers your easy questions! Or indeed anything to do with *Grub, LILO, netatalk, vi...*



Hans Huberland is Rackspace Managed Hosting's Linux expert. Send any Linux system admin questions to **sysadminqa@rackspace.co.uk**

Slack off

Q Encouraged by your recent article on the Slackware distro, we tried it out and then bought a boxed set to use alongside SUSE 9. We like it but have immense problems with its 'permissions' policy and would like your help on this.

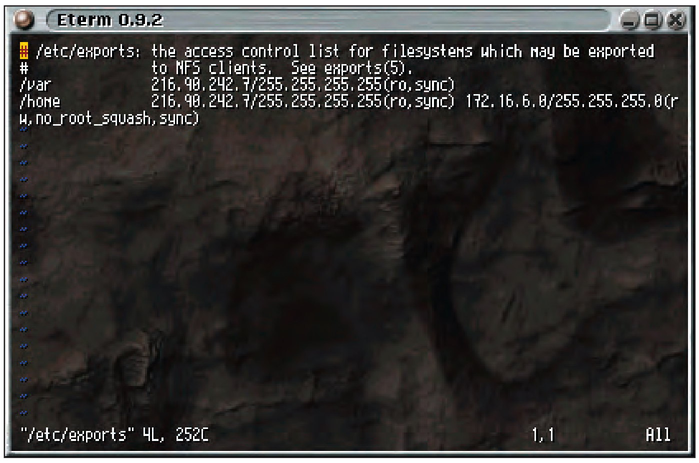
We set *umark* at 0022 as usual but find the following:

1 *NFS* works well in transferring directories between machines running Slackware via our LAN, but they are unusable once they get there as we always get the message *"you do not have permission ..."* This despite the username and password being the same on both machines. (We have no problems between machines running SUSE). Occasionally, I even managed to access my son's files without his permission, perhaps because he had so much trouble with sound that he removed root privileges from everything he could access? I am able to get limited access from Slackware to SUSE using *Samba,* but it is one-way only. We can, of course, send data between the machines via my website, but this seems a bizarre method to adopt.

2 Creating any file or directory in one's own space ends up with a file/directory with root/root permissions and so needs *chown* and *chgrp* to correct it. Again this seems unusual and does not fit with the usual instruction of never using root for routine operations.

We installed Slackware as per the instructions in its 'essentials' manual, but can find no clues in it as to how to correct this anomaly.

Joe Lamb, *via email*



```
Eterm 0.9.2
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
/var            216.90.242.7/255.255.255.255(ro,sync)
/home           216.90.242.7/255.255.255.255(ro,sync) 172.16.6.0/255.255.255.0(r
w,no_root_squash,sync)
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/exports" 4L, 252C                                        1,1           All
```

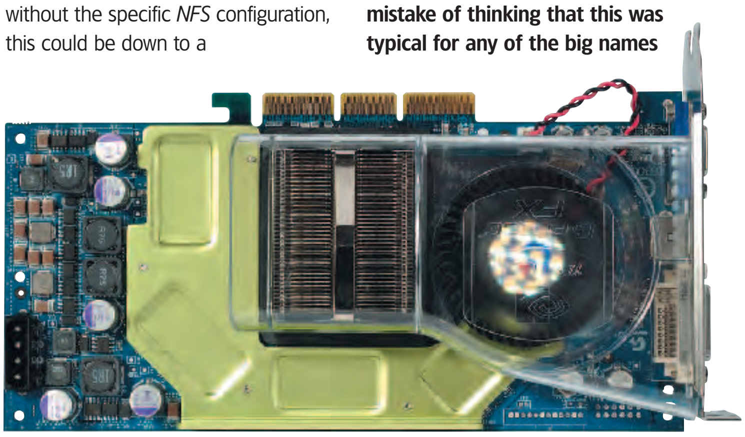
***NFS* has a number of options for manipulating UID/GID information for users, however it's generally assumed that the same users have the same UIDs on all systems**

A All Unix permissions are based upon the UID and GID, rather than the specific username of the user. It is quite possible to have two systems – both with user accounts using the same name – but with different UIDs limiting access over services such as *NFS*. If you log into each system using your individual user accounts, the **id** command will show you which UID you have on each system. Using a umask of **0022** will allow files to be created with world-read privileges, so independent of the *NFS* issues, they will be accessible by any user.

It's unusual that you creating files in a directory via *NFS* would yield root/root permissions, although without the specific *NFS* configuration, this could be down to a misconfiguration on the *NFS* side of things, or if it is a permissions issue on the individual systems.

Godawful graphics

Q A while back, my graphics card started bombing out occasionally, so I decided to treat myself to a newer, improved model. My old card was a generic Nvidia GeForce 4 card and worked perfectly under both Windows XP and Linux. I suppose that because the Nvidia drivers had worked so well under Linux, I made the mistake of thinking that this was typical for any of the big names



Gigabyte GeForce FX5950 Ultra works under Linux – see *LXF53* for more.

in graphics cards. How wrong I was! The new card I bought was a Powercolor Radeon 9600XT. It cost just over £100 and, I thought, was a good buy. However, I just cannot believe that getting this card to work in Linux (I'm using the Mandrake 10 release from last month's *LXF* cover-mounted DVD) would be so painful. I've spent what seems like a lifetime crawling around the Internet, looking for HOWTOs– and have tried many – but with no luck.

I've included various files (*XFree86.0.log, Xfree86Config-4, glxgears output, glxinfo*) and the text from the HOWTO which I followed. By commenting out the Load dri line near the beginning of *XFree86Config-4*, X will start, but I find that 3D support is missing. Also, the *XFree86.0.log* this creates is strewn with errors. If, however, I don't comment the line out, then I get a black screen which I can only get rid of by pushing the reset button. Strangely enough though, the *XFree86.0.log* that this produces is error-free.

After having spent so much time searching the Internet for clues, I can see that "ATI" and "good driver support" are words that are rarely – if ever – found in the same sentence. From my point-of-view, they really should be black-listed, never mind ashamed!

Anyway, moaning apart, I'm sure that there must be hundreds of *LXF* readers who are in the same boat as me, all of whom would be grateful for your advice. Maybe you could do a detailed HOWTO of your own in a forthcoming issues?

As for me, I promise that I will never buy another ATI product!

Mick Scully, *via email*

A Nvidia is an exception in the Linux world, as it distributes binary-only drivers for its video chipsets, rather than requiring someone to reverse-engineer the drivers by little more than trial-and-error, or using whatever incomplete list of specifications they can wheedle out of the manufacturer.

As the device functions without DRI support, it would suggest an issue with the drivers. Ensuring that you have the most up to date XFree86 installation available for your distribution will negate any issues with known bugs.

Depending upon the manufacturer and their support of Open Source, 3D support in Linux can be anything between non-existent through to complete OpenGL functionality. As popular as ATI is, the support for its devices under Linux can be fairly patchy. It may very well be that there is a future release of XFree86 that solves many problems that people have with ATI, but as ATI does not contribute to the project, all support must be built by third-parties.

Kpackage

Q What has happened to *Kpackage*? My concern is not that it is missing from Mandrake 10.0 (I can get it from elsewhere) but the implications of the absence for its development and support.

The problem is that, in spite of its limitations, *Kpackage* is unique: I use it as my file manager because it lists files by package (for normal file operations I use the command-line). Few packages come with a 'getting started' Readme, and the install script splatters files all over the system. A quick look at *Kpackage* shows you not only where the files are, but also what their names are; and you do not get very far with Linux unless you know the names of executables, config files, documentation *etc*.

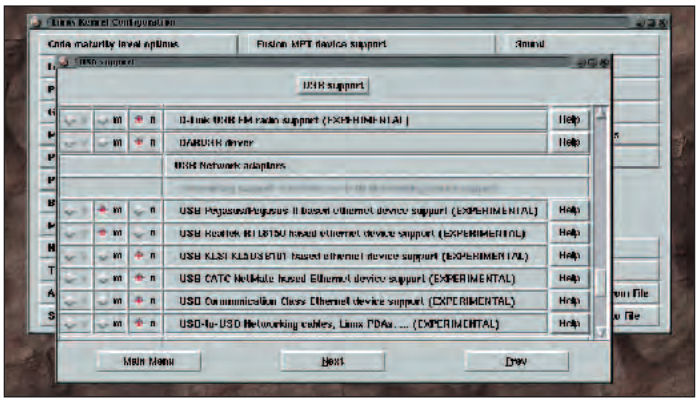
It is thanks to *Kpackage* that I was able to find my way around Mandrake when I first started using it; conversely, I have hit a brick wall with Debian because *Kpackage* is either missing or empty.

The information which *Kpackage* provides is available from RPM, but it is the presentation which makes it useful: do you know of any other program which does a better job of showing this information?

Cecil Wallis, *via email*

A Mandrake has its own package management tool, although it does not have the extensive functionality that *Kpackage* has. We're not aware of any other tool that works as a file manager, making *Kpackage* fairly unique. Of course, there is nothing stopping you from using *Kpackage* with Mandrake by downloading the RPMs. As *Kpackage* is specific to Linux distributions using RPMs, it probably won't be as widely developed as other components of KDE; however that doesn't mean someone out there is not continually maintaining it.

Kpackage will not be a whole lot of use with Debian, as Debian does not use RPMs for package management, instead using *dpkg*.



Linux Kernel Configuration

Code maturity level options | Fusion MPT device support | Sound

USB support

USB support

○ y ○ m ● n D-Link USB FM radio support (EXPERIMENTAL) Help
○ y ○ m ● n DABUSB driver Help
USB Network adaptors
○ y ● m ○ n USB Pegasus/Pegasus II based ethernet device support (EXPERIMENTAL) Help
○ y ● m ○ n USB Realtek RTL8150 based ethernet device support (EXPERIMENTAL) Help
○ y ○ m ● n USB KLSI KL5USB101 based ethernet device support (EXPERIMENTAL) Help
○ y ○ m ● n USB CATC NetMate based Ethernet device support (EXPERIMENTAL) Help
○ y ○ m ● n USB Communication Class Ethernet device support (EXPERIMENTAL) Help
○ y ○ m ● n USB-to-USB Networking cables, Linux PDAs, ... (EXPERIMENTAL) Help

Main Menu | Next | Prev

Linux has support for a wide variety of USB network devices, however one needs to know which chipset is required in order to build the appropriate kernel module.

NICs nicked?

Q I greatly enjoyed your article in *LXF53* about making hardware work under Linux. Unfortunately, I noticed that even though you covered USB, you didn't talk about USB NICs, and even though you talked about Modems, you didn't talk about USB based modems.

So, as a person who compiles his kernel from source, I have yet to figure out what module adds USB NIC functionality. Do you have any pointers?

David Li, *via email*

A There are a number of different chipsets used by USB Network Interfaces, although the vast majority make use of the 'Pegasus2' chipset. The simplest method is to plug the device in and run *dmesg*, which will provide information, including the vendor and device ID through *dmesg*, which can be fed into Google or **www.linux-usb.org/** to identify the specific kernel module which needs to be loaded. There are a number of different modules available within the USB section of the kernel, so compiling all of them, loading them all, then plugging the device in will identify which kernel module it requires.

Install ire

Q I am new to Linux – I just migrated using the OS after purchasing *LXF51* for the first time with its DVD containing eight Mini distros. I am having some difficulties installing ASP, Fedora, Mandrake, SUSE and Red Hat on my Windows XP platform using *Partition Expert*.

I am not sure, but at first installation using partition expert I resized my XP as FAT32 primary, and then created the below:

WindowsXP	Fat32	Primary
/boot	x3	logical
Linux swap	x3	logical
asp	x3	logical
fedora	x3	logical
mandrake	x3	logical
suse	x3	logical
redhat	x3	logical

Is there anything I am doing wrong pertaining to installation? My systems refused to allow me to choose which OS I prefer to use. Can you kindly advice me on how to successfully install the operating systems mentioned above?

Tun Bosun, *via email*

A The first step would be to remove the /boot filesystem, as you generally don't need one anymore. Each distribution is going to require its own kernel and boot process. However, you may want to share /home across all of you installed distributions, so that you can keep a consistent home directory, no matter which you boot into. You will also have to use *swap* as the type for your shared swap filesystem, rather than ext3.

You should be able to install each distribution onto the individual filesystems, although maintaining *LILO* or *Grub* is no doubt going to be quite a chore. As always, Windows should be the first thing you install, as it always has a wonderful tendency of blowing away everything else.

»

A HELPING HAND WITH... MANDRAKE 9.1, 9.2 & 10.0

Unsound MDK

I installed Mandrake 10.0 relatively easily but am having great difficulty configuring the sound card, modem and printers. Any ideas how to overcome this? Your cover DVD will not now load in my 'hdc' drive because it is a CD-ROM type, my DVD/CD-ROM is now configured as 'hdd'. How do I change them over so I can access files on the coverdisc?
David Day, *via email*

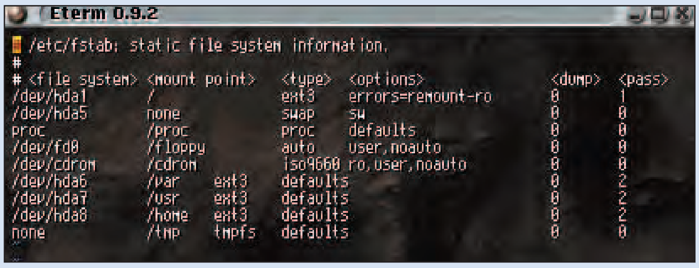
Definitions for mounting devices on a specific directory can be found in /etc/fstab, which you will have to edit as root. An example fstab entry for mounting a CD is below.

```
/dev/cdrom    /cdrom   iso9660
ro,user,noauto     0       0
```

Your fstab file may either identify /dev/hdc directly, or access it through a /dev/cdrom symlink. In the first case, it is as simple as changing the **hdc** to **hdd**. For a symlink, you can delete the existing /dev/cdrom and run:

```
# ln -sf /dev/hdd /dev/cdrom
```

Some applications that access DVDs – such as *Xine* – will look at /dev/dvd too, so ensuring that both references are modified will avoid any future occurrences of the system not being able to mount or access a CD or DVD.



```
/etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>                <dump>  <pass>
/dev/hda1       /               ext3    errors=remount-ro        0       1
/dev/hda5       none            swap    sw                       0       0
proc            /proc           proc    defaults                 0       0
/dev/fd0        /floppy         auto    user,noauto              0       0
/dev/cdrom      /cdrom          iso9660 ro,user,noauto           0       0
/dev/hda6       /var     ext3   defaults                         0       2
/dev/hda7       /usr     ext3   defaults                         0       2
/dev/hda8       /home    ext3   defaults                         0       2
none            /tmp     tmpfs  defaults                         0       0
```

There are a number of different ways through which a CD device can be referenced, but using symlinks keeps everything neat.

I currently use Mandrake 9.1 and was eagerly waiting for the release of Mandrake 10.0, having been unable to install 9.2. The installation went smoothly, but then when I tried to reboot and startKDE properly, I was faced with a message saying that the sound server was unable to start up, and would therefore continue to use the null sound output. Result? No sound! I then went into *Mandrake Control Center>HardDrake>* Soundcard>Configure and chose the appropriate driver. It configured correctly but there were still problems: chiefly, no system notification sounds.

It did not stop there however, for when I shutdown and then rebooted the system later on, everything had gone back to how it was before, and I was presented with that very same message all over again. Please help, as I would like to move up to Mandrake 10, but I can't do so until this is resolved.

May I suggest that you inform MandrakeSoft itself of this apparent bug? My computer is a Pentium 4 1.7GHz, and my soundcard is a Sound Blaster Live 5.1. Thank you.
David Kendall, *via email*

Mandrake 10 is based around the Linux 2.6 kernel, so any issues which you have with Mandrake 9.2 may be solved, and indeed, you may have many more changes and issues. It sounds more like a problem with the system loading the correct modules at boot time, rather than anything else. If you perform the upgrade to Mandrake 10, it may clean up this issue. Without going further and investigating the reason why the kernel modules are not loading, or if they are, work out exactly why the sound device can not be accessed. Running *dmesg* at boot time will indicate if the modules actually loaded correctly, and if they did, you can investigate if the /dev entries are inaccessible by the user which attempts to access the sound card.

Of course, moving to Mandrake 10 may be a good reason to do a clean-up; and rather than do an upgrade, do a clean install. Doing an upgrade is never quite the same as doing a fresh install on the box.

Panic!

I'm a complete newbie to Linux. This week I created an 8GB Linux partition and installed Mandrake 9.2 alongside Windows 98SE. Everything has gone swimmingly – no problems at all. It seems to have detected all my hardware devices OK as well, and the bootloader works too. Then, I visited your website and was alarmed to see your message about *serious problems* with the Mandrake distribution I've used, including possible hardware damage. Does the fact that everything is OK now with my installation mean I've been lucky, or do I need to take some remedial action? Should I continue exploring my new OS or should I ditch it? What signs of trouble should I watch out for?
Martin Barge.

It is highly unlikely that any operating system can cause hardware damage, although it's not unusual for a kernel to destroy data. We are yet to see any operating system destroy a system to the point where it can not be used. As Mandrake 9.2 is a minor revision of Mandrake 9, it should not cause any major problems. Usually it is the distribution versions ending in zero that cause the most problems, such as Mandrake 9.0 and 10.0.

Everyone's hardware is different, so it may be that Mandrake 9.2 has issues with a specific IDE controller, SCSI device or something in between. If you've not had problems and it's running happily, then there probably is little to worry about. I a

Laptop MDK

I just installed Mandrake 10.0 on a new Averatec laptop. Everything worked right from the start except USB, which is one of the few things to ever not have given me trouble in the past. I'll list everything I've found out and tried below.

First, I get these relevant messages during boot.

```
Starting USB Host Controller USB-UHCI
Starting USB Host Controller EHCI-HCD
Mounting USB File System
```

***Mandrake Control Center* has these listings under USB Controllers. /dev/usb has an empty folder**

« One has to ask the question why all these distributions need to be installed. Rather than battle to install six at once, maybe installing one at a time and trying them out for a week or two would be a better way to test how distributions really function.

Hot, hot hotkeys

Q I have spent the last few years trying to convince myself that Linux is ready for MY desktop, and with the release of Mandrake 10.0 Community Edition, I think I've finally made the switch for good. My goal now is to get everything working perfectly on my system, including my keyboard hotkeys. Your *Ultimate Linux Box* article has been a really great help with this but I have now got a problem.

I have got all the hotkeys working fine except for the volume up, down and mute keys. I can't work a way to control these from the KDE keyboard shortcuts dialog. I think that I need to use aumix as I can control the volume using a command such as:

```
aumix -v +10
```

but I can't find a way to assign that command – with the command-line arguments – to a hotkey.
Matt Fletcher, *Stanley Road Baptist Church*

A Hotkeys on keyboards must first be defined before they can have a function applied to do them. Usually, one will assign **F13**, **F14**, and so forth to keys which exist on the system.

If we create a file called ~/.Xmodmap and have its contents be:

```
keycode 115=F13
keycode 116=F14
keycode 117=F15
```

To find the actual number, such as 115, one can use *xev* to test the keys on the keyboard and find out exactly what code they pass to X. Once this has been done, you can use *xmodmap* to load the new configuration then use KDE to setup the mappings to have them execute **aumix** or any other command you wish.

named 'hid'. USBView gives the message: 'Cannot open the file /proc/bus/usb/devices. Make sure you have USB compiled into your kernel, have the USB core loaded, and have the usbdevfs filesystem mounted.'

lsmod gives these relevant entries:

```
usb-storage   63136  0
scsi_mod      114744  6 ppa,imm,
sd_mod,sg,sr_mod,usb-storage
ehci-hcd      24196  0
uhci-hcd      29104  0
usbcore       99132  5 hid,usb-
storage,ehci-hcd,uhci-hcd
```

modprobe -vv usb-uhci replies:

```
install /sbin/modprobe uhci-hcd
```

After inserting a Memorex USB 128MB flash memory card, dmesg | grep usb reports:

```
drivers/usb/core/usb.c: registered
new driver usbfs
drivers/usb/core/usb.c: registered
new driver hub
drivers/usb/host/uhci-hcd.c: USB
Universal Host Controller Interface
driver v2.1
drivers/usb/core/hcd-pci.c: Found HC
with no IRQ.  Check BIOS/PCI
0000:00:10.0 setup!
drivers/usb/core/hcd-pci.c: Found HC
with no IRQ.  Check BIOS/PCI
0000:00:10.1 setup!
drivers/usb/core/hcd-pci.c: Found HC
with no IRQ.  Check BIOS/PCI
0000:00:10.2 setup!
drivers/usb/core/hcd-pci.c: Found HC
with no IRQ.  Check BIOS/PCI
0000:00:10.3 setup!
drivers/usb/core/usb.c: registered
new driver usb-storage
drivers/usb/core/usb.c: registered
new driver hiddev
drivers/usb/core/usb.c: registered
new driver hid
drivers/usb/input/hid-core.c:
v2.0:USB HID core driver
```

Note:The BIOS setup utility does not have a PCI or IRQ setup option. Here's /etc/fstab

```
/dev/hda5 / ext3 noatime 1 1
/dev/hda1 /Documents ext3
user,dev,noatime,suid,exec 1 2
none /dev/pts devpts mode=0620 0 0
/dev/hda9 /home ext3 noatime 1 2
/dev/hdc /mnt/cdrom auto umask=
0022,user,iocharset=iso8859-
1,codepage=850,noauto,ro,exec 0 0
none /proc proc defaults 0 0
none /tmp tmpfs defaults 0 0
/dev/hda7 /usr ext3 noatime 1 2
/dev/hda8 /var ext3 noatime 1 2
/dev/hda6 swap swap defaults 0 0
/proc/bus/usb /proc/bus/usb usbdevfs
defaults 0 0
```

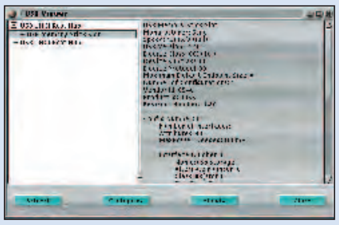
No USB device seems to be recognised. The light on the Memorex card comes on during the boot process, so I believe the USB hardware works, but it turns off when *LILO* comes up, so I'm sure it's a driver or module issue, but I can't figure it out.

***Sam Hogue,** Lugoff, SC USA*

The first step should be to boot the system with the **pci=biosirq** flag, which modifies the way IRQs are handled by the system. It would appear that the system has a functional USB system, however it doesn't know how to allocate IRQs to devices when you connect them.

We would first suggest looking in /proc/bus/usb and verifying that the 'devices' file does not exist. One needs to be root in order to access USB information in /proc, so running **usbview** as root or through **sudo** is the only way for **usbview** to open those files.

The complete output from **dmesg** would be very useful, as at this stage we don't know if a SCSI device was registered with the system, or if the storage medium was completely ignored. The USB storage device will be allocated a standard SCSI device, such as **sda**, **sdb** and so forth. The only time /dev/usb/ is used is for devices such as serial ports, printers and other such devices which are not part of an existing subsystem.



***usbview* uses /proc/bus/usb to look at the specific devices connected to a USB bus, however one needs to be root to read the files.**

rackspace MANAGED HOSTING

Linux ISPs?

Q Hi, I'm currently struggling to find an ISP that is compatible with Linux. Can you please advise me of some options where I can use Red Hat or maybe Free BSD and surf online via broadband? Thanks!
Gandalf, *via magic*

A Almost every ISP we've come across at *Linux Format* or at Rackspace is certainly Linux 'compatible' (though they may not provide telephone support for Linux). In essence, every Internet connection operates on a network level. So, if Linux speaks the correct protocol (which it usually does) you should be able to get onto the Internet.

What's more likely to cause a problem is getting a USB ADSL modem that works. I recommend looking online to check if Linux supports the device or if there are third-party drivers available.

Alternatively, a more popular and far easier option would be to get hold of an ADSL router. You would generally configure an ADSL router from a web page, and it would automatically assign the correct IP settings to your Linux system.

If you're still looking for advice on an ADSL ISP, I would highly recommend doing some research on **www.adslguide.org** – here you can compare ISPs based on customer ratings for speed, reliability and customer service.

Gentoo networking

Q I've been using Linux for several years and have switched distributions regularly. I always find that each new distro I try has something about it which really annoys me. I recently tried Gentoo Linux, and although there is quite a bit of work involved in getting it loaded, I love the fact that I can turn it into whatever I want. I just installed Gentoo 2004.00 onto my laptop and so far so good. I have only one issue but I'm sure you'll be able to point me in the right direction here.

As this is a laptop, I do not always have a network cable plugged in. Gentoo will try to get *eth0* up and will not continue booting until *DHCP* has timed out. I don't know exactly how long this takes but it sure feels like at least minute or two – quite annoying when a normal bootup only takes about 30 seconds.
Jason, *via email*

A We have heard of this before – sadly, the problem you're experiencing is not easy to solve, but there is a very easy work-around. Gentoo's baselayout package contains the scripts that govern network startup. They would be pretty tricky to edit manually to solve your problem and next time Gentoo updates this script you would lose your changes unless you carefully reviewed which files you were updating.

As a workaround, I would recommend installing *ifplugd*. This is a hotplugging daemon for network interfaces that watches your network card and will bring it up or down as soon as a cable is plugged in or is unplugged. It will not attempt to start *eth0* unless it can see a link on that interface.

To install it simply

```
emerge ifplugd
```

It's most useful when it is started at each bootup, to do this simply enter:

```
rc-update add ifplugd default
```

Registrar

Q I have a problem with my domain name. I have just moved to Rackspace and added the domain to your nameservers. I was told that it could take 24 hours or so for the DNS to propagate but it's been over a week now and it still points to my old hosting provider. Am I missing something? How can I find out where the problem is?
Cathy, *via email*

A Hello Cathy and welcome to Rackspace! I believe the problem you are having is due to your registrar not being aware that the name servers for you domain have changed. A very easy way to verify this is by using the **whois** command. From your linux command-line, type:

```
whois domain.com
```

Near the bottom of the output you should find some information relating to the name servers for this domain. If you find that these are still pointing to your previous provider then this would explain why your results are what they are. You should contact your registrar to update these if they are incorrect. Depending on your registrar, this is probably done through a secure customer web interface. Remember that even once this has been changed, you will still need to wait for propagation to take place and this can take 24 hours (sometimes more).

File transfer

Q More and more often, I have had issues in finding a way to transfer files to fellow sysadmins or clients. The issue starts me with being behind a corporate router, so an FTP server on my computer won't be accessible.

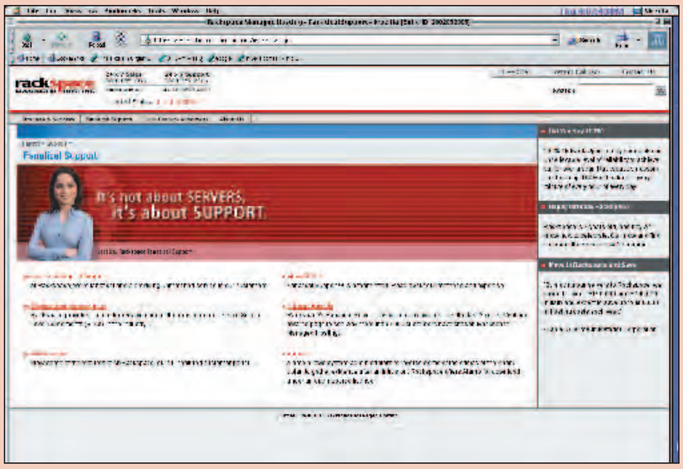
Also, our ISPs aren't happy when we send 100MB+ files to each other's mailboxes. For a while, I have successfully managed to use www.files2U.com, but now it unfortunately wants to start charging for its services. I was thinking that the receiver could have an FTP server instead and I can put the file to their server. I am wondering if there is a way for me to send files to others with minimal effort for the receiver.
David, *via email*

A The problem you are faced with is a difficult one. With security being such a huge issue on the Internet, most people do not allow inbound connections to

ALL ABOUT RACKSPACE

When it comes to managed hosting Rackspace is unique. We are the only hosting company to guarantee 100 per cent network uptime and we will even pledge to replace faulty hardware (let's face it, it does happen) within an hour.

As you would expect from Red Hat's only Advanced Hosting partner you will receive Fanatical Support™ and instant emergency response from fully qualified level three technicians, available 24-hours a day, 365 days a year. No answering machines, no diversions, no silly on-hold messages. All our customer platforms are housed in state-of-the-art, secure data centres and can be fully customised to meet any requirement. A dedicated account manager is allocated to you at the outset and will remain on your account. All Rackspace employees are committed to ensuring customer satisfaction, and will not rest until a client problem is resolved. Since Rackspace was established in 1998, it has specialised in managed hosting – nothing else. We're not an ISP, we don't sell domain names, and we won't offer website design – all we do is managed hosting. As a result, we do it very well and 97 per cent of our customers would happily recommend us. For information on how we can improve your web hosting, please contact us at www.rackspace.co.uk and ask any questions you may have about the ultimate managed hosting service.



WIN an Archos AV140 Video Recorder WITH RACKSPACE MANAGED HOSTING

www.rackspace.co.uk

Every month, the best question related to Systems Administration that a *LXF* reader sends in wins a prize: this month, you have the chance to win the awesome Archos AV140 Video Recorder.

The AV140 combines a modular MP4 video player and recorder, MP3 music player and recorder, digital camera and camcorder, digital photo wallet and data storage in a compact device that fits in the palm of your hand and is compatible with Linux. Based on a 40GB hard disk that is completely compatible with Linux, Windows and Mac OS, all you have to do is connect it to your computer and it will be mounted as an additional hard disk – on which you can store *any* type of data. USB 2.0 auto-connection ensures fast data transfer to and from your PC.

You can now enjoy 160 hours of near-DVD quality MP4 video with MP3 sound wherever you are. You can record and playback to your TV, VCR or camcorder directly in MP4 format. Playback is an AVI format file with MPEG4-SP VHS quality video and MP3 stereo sound, with the ability to record up to 2,000 hours of voice audio. See www.archos.com/products/prw_500432_specs.html.

Email all your sysadmin questions and problems to our expert system administrator Hans Huberland: sysadminqa@rackspace.co.uk

Every published query on these system admin pages wins a Rackspace T-shirt!

★Star Question – AV140 winner!

This issue's lucky winner is ***Richard*** – your prize will be with you shortly!

Q I thoroughly enjoy your magazine. It is well written and the layout is very appealing to the eye. The question that I pose for your *Answers* column is this: I presently have two hard drives in my computer (both the same make and model). I would like to know what would be considered an optimal partition setup for Linux so that the workload is equally distributed between drives, and computer performance and security of files is enhanced.

What filesystem formats should be considered, and where should mount points be placed (Primary Drive, and Extended Drive)?

(ie file system formats: Reiser, Ext2,Ext3,Swap. Mount Points: /boot,/var,/opt,/home,/srv,/local.)
Richard, *via email*

A Thank you for your flattery, Richard. Looking at the *LXF* forums, you'd think – from some posters' point-of-view – that we draw the magazines in green crayon on the back of old crisp packets! To answer your question, there is no single ideal configuration, for both filesystem types and mount point layout. However, understanding the pros and cons of the various options will help you make an informed choice.

ext2 and ext3 are actually different versions of the exact same filesystem, with the exception that ext3 has journaling support. Journaling adds metadata to the filesystem so that if any inconstancies are found during boot up they can be repaired much faster. We can't think of any compelling reason why anybody would choose the older ext2 over ext3, so if your distribution supports it (and almost all do these days) I would recommend it as an option.

ReiserFS is another journaling filesystem which is considered to be very fast and reliable. It works phenomenally with a large quantity of small files, but handles larger files very well too. Another lesser known journaling filesystem is XFS. It does a lot of caching of data and is therefore only recommended on very fast hard drive systems (SCSI or fibre channel) with UPS power available.

Keeping your data over many partitions offers no performance benefits. The more filesystems/partitions that your data is spread over, the easier it is to backup or restore data. On the down-side, you need to plan the use of systems far more carefully if you would like to use many partitions. If you run out of space on a partition, you cannot easily grow.

With servers most people will create a /boot, a / and sometimes a /home and /var (if there is significant data there). In a workstation environment, just a / and /boot are required. As you have a second drive I would either use it for /home (if you will be keeping your personal data there) or /data. Different distributions use different locations for user installed applications, some use /usr and other use /opt. If you are going to be installing a lot of applications then it may be worthwhile creating one of these too.

One final point: as you have two hard drives, I would highly recommend splitting your swap space over both these drives. To do this, simply create a swap partition on both drives. If swap space is used, it will spread the workload over both disk controllers in this case.

workstations anymore. Even connections to servers are being tightened down. I fear that even if your receiver sets up an FTP server on their system that any firewall on their network will block your attempted connection anyway. At the very least it will probably not be a reliable method.

The only definite way to do this is to have some storage space available on a server that explicitly allows that type of connection into it (such as files2U). Unfortunately most of the cost of using the internet comes from the cost of bandwidth, this means that you may have a hard time finding a supplier who is willing to let you upload a file of 100Mb and have it downloaded on a regular basis. Your only choice may be to use a paid for service. Files2U is one option but a simple dedicated ftp server or an account on a server with some webspace should work.

« AtomicTanks

Q I am very new to Linux but want to give it a go. I have Mandrake 9.2 dual-booted with Windows98. All the hardware – CD drives, Modem and Printer – have been installed without any problems, and I've even managed to load a RPM package. My big problem is with tarballs, in particular *AtomicTanks*. As my daughter plays the BBC Acorn version, I thought I'd try to install it on my machine from *LXF52*'s DVD. I managed to unzip the game using *Konsole* in root by issuing tar xzvf /mnt/cdrom/<file name> followed by cd <file name>, then make install . I now have a folder in my Home directory called *Atanks* full of sub-folders and text files, but I am unable to play the game – am I missing something? Typing cd atanks, then ./atanks from *Konsole* only produces the message bash: ./atanks: No such file or directory. I hope that I've provided enough info to resolve the problem.

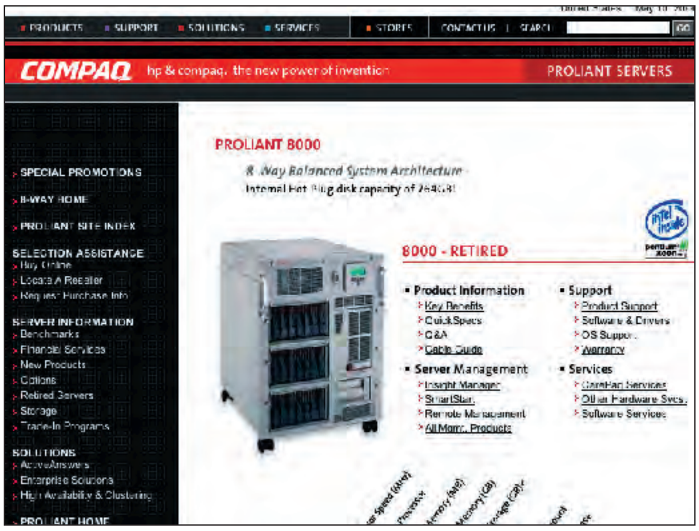
Richard Barlow, *via email*

A When you issue the **make install** command, it will install the program and supporting files in a specific location on your system, usually under /usr/local, which you will have to do as root. However, you should be able to run it from the unpacked tarball without having to do anything too exciting. The error you received implied that the *atanks* binary does not exist, so the first step would be to use **ls** to look in the directory and see if the file exists. It may be that there is an **src** sub-directory which contains the actual binary, rather than having it within the top-level directory that is created by the tarball.

There is generally a README file that is contained within the top-level directory, which should explain which binaries are installed, and where they should go.

Posting to the forum
The *LXF* online community

Not only do our popular forums at www.linuxformat.co.uk have sections dedicated to your technical queries, hardware, programming languages and general help; but also there's always a lively Linux discussion going on!



Even though a piece of kit is 'retired,' you will still be able to find a lot of information about it on the Internet: either at manufacturers' sites, as is the case with this Compaq server, or with a bit of judicious Googling.

Any old iron?

Q I purchased a copy of your magazine, as I was interested in setting up a Linux server for home use, but I'm having problems with the install. I managed to obtain an old – circa 1998 – Compaq Proliant 8000 with a RAID 5 array, that was previously configured as a Netware 4.11 server. I can't get the installation to mount the file system, also having trouble with memory, even though the system has over half a gigabyte of memory, can get round this with the option:

```
linux mem=512M noprobe
```

I would appreciate your thoughts, as I really need to know how to wipe the RAID 5 partition, and/or create the required 'linux' partition as one logical drive.

Mike Greenwood, *via email*

A It's always great to hear from readers who are recycling legacy technology! You will be able to partition the RAID array, as if it was any other storage medium. It does depend which RAID controller you have on the box, as the Linux kernel you boot into will have to support it. You've not said which distributions you've attempted to boot the system into, although most success can be had with Red Hat or Fedora, as they generally have support for every piece of hardware under the sun. If you boot from a Red Hat CD and then switch to a command line, you will be able to check what devices the installer picks up and look at the partition table, if any, on the RAID array.

Something's Mything

Q *LXF51*'s DVD contained *MythTV*, and it was commented on further in *LXF52*. It seemed like a good idea, so I spent some time getting all the needed software installed on my 'Yarrow' Fedora Installation with a Radeon 7000 64MB Video Card and a Wonder Pro TV card (by Z cyber) all living happily on my P4 Asus motherboard running a 2.2GHz Celeron processor.

After following the instructions, *ie* installing the Perl Modules and various and sundry libraries and *MySQL* tampering, I arrived at the *MythTV* directory itself (near mount *Doom*) at the final setup/setup instruction this simple error floored me after running setup:

```
Session management error:
Authentication Rejected, reason :
None of the authentication protocols
 specified are supported and host-
based authentication failed
QSqlDatabase warning: QMYSQL3
driver not loaded
QSqlDatabase: available drivers:
QSqlDatabase warning: QMYSQL3
driver not loaded
QSqlDatabase: available drivers:
Unable to connect to database!
No error type from QSqlError?
Unable to open database:
```

Driver error was:

```
Driver not loaded
```

Database error was:

```
Driver not loaded
```

This driver/module is available in the QT enterprise development kit which I do not have?

Peter Retief, *via email*

A From the information you've provided, it looks like *Qt* is unable to load the *MySQL* driver, which can be solved by adding **/usr/qt/3/plugins** to **library paths** using *qtconfig*. You should also verify that the *MySQL* driver for *Qt* is not included as part of a separate package, rather than along with the rest of the *Qt* system. LXF

For more on the ins and outs of using *MythTV*, please see *LXF52* – order your backissues on page 94.

Submission advice

We are happy to answer all sorts of Linux-related questions. If we don't know the answer, we'll find out for you! But in order to give you the best service, it helps a lot if you read the following submission advice.

- Please be sure to include any relevant details of your system. *"I can't get Z to work"* doesn't really mean anything to us if we don't know things like what version of Z you are trying to run, what hardware you are running on.
- Be specific about your problem. Things like *"it doesn't work"* or *"I get an error"* aren't all that helpful. In what way does something not work? What were you expecting to happen? What does the error message actually say?
- Please remember that the people who write this magazine are NOT the authors or developers of Linux, any particular package or distro. Sometimes the people responsible for software have more information available on websites *etc.* Try reading the documentation!

We will try to answer most questions. If we don't answer yours specifically, you'll probably find we've answered one just like it. We regret that we can't give personal replies to all your questions.

WRITE TO US AT:
***Linux Format*, Future Publishing, 30 Monmouth Street, Bath BA1 2BW** or
email: **lxf.answers@futurenet.co.uk**

missed one?

LINUX FORMAT BACK ISSUES

Every month *Linux Format* brings you the best tutorials, the essential reviews and the latest news. But if you've missed out on a must-read feature or a vital program from our expertly compiled CDs and DVDs, order your back issue NOW! And remember, you need never miss an issue of your favourite Linux mag, if you subscribe to *Linux Format* (see overleaf for more details).

June 2004

Product code:
LXFB0055(cd)
LXFDB0055(dvd)

MAGAZINE FEATURING:
GET PROTECTED- combating your PC problems, Creative Commons, VMware GSX, Audacity, Java 2, TuxCards, Pedora Core 2, SUSE Pro 9.1, KDevelop tutorial: Part one

DVD HIGHLIGHTS:
Live Distro Bonanza! Seven ready to burn – including MandrakeMove, plus Knoppix boots directly from the DVD

CDs HIGHLIGHTS:
Knoppix, Damn Small Linux, Slax, Storix Personal, RPM Linux, GNOME 2.6, Security applications, The GIMP 2

May 2004

Product code:
LXFB0053(cd)
LXFDB0053(dvd)

MAGAZINE FEATURING:
Build the ULTIMATE Linux PC, Mandrake 10.0, C/C++ IDEs, Tim O'Reilly exclusive interview, UT2004, SQL databases, Mono, Evolution, Subversion, GIMP 2 preview, KDE 3.2

DVD HIGHLIGHTS:
Komics, K3b, Unison, Unreal Tournament 2004, Subversion

CDs HIGHLIGHTS:
Mandrake 10.0 (including Kernel 2.6.3, KDE 3.2, Mozilla 1.6, OpenOffice.org 1.1.0, GIMP 2, Samba 3.0, MySQL4 etc) Kino, Subversion, NET-SNMP

April 2004

Product code:
LXFB0052(cd)
LXFDB0052(dvd)

MAGAZINE FEATURING:
Making movies, Opteron or Itanium? Compression utilities, HDL, SDL game programming tutorial, Perl Template Toolkit, , KOrganizer, Eric Raymond, SNMP, Arkeia 5.2 review

DVD HIGHLIGHTS:
8 Distros, Gcompris, MyPasswordSafe, SystemRescueCD

CDs HIGHLIGHTS:
Tons of video apps, KDE 3.2, ClarkConnect, SMEServer, LMsensors, hdparm, LiarLiar, QTVision, SpamAssassin, AtomicTanks, ifplugd, Roundup51

March 2004

Product code:
LXFB0051(cd)
LXFDB0051(dvd)

MAGAZINE FEATURING:
Get your hardware working, Tadpole laptop, Xandros OS2, Firewalls Roundup, PixiePlus, Armari 4-way Opteron, Xinit SPS440, Wine Rack, Opteron vs Itanium2, MandrakeMove

DVD HIGHLIGHTS:
Java Desktop System, 8 mini distros, Scrubber, Python 2.3.3, Bacula

CDs HIGHLIGHTS:
MandrakeMove, Seapine Surround SCM 2.1, Simple CDR-X, MythTV, Six mini distros, Konserve, Tuxpaint, Kernels 2.4.24 & 2.6.1, Hardware applications

February 2004

Product code:
LXFB0050(cd)
LXFDB0050(dvd)

MAGAZINE FEATURING:
Big Hitters biog collection, Extreme Programming Pros & Cons, FORTRAN, CrossOver Office, Systemax 6507 & Evesham Reliance servers, Astronomy apps, Blender

DVD HIGHLIGHTS:
EVERY SINGLE HOTPICK EVER! All the programs that have featured in *LXF*'s Hot Picks selection from issues 1 to 50

CDs HIGHLIGHTS:
50 HOTTEST PICKS! Including: Anjuta, ZNES, Planner, Scribus, Mozilla, Anjuta, Exim, ProFTPD...

January 2004

Product code:
LXFB0049(cd)
LXFDB0049(dvd)

MAGAZINE FEATURING:
Java Desktop Environment, Linux DVD Players, CD copy protection feature, Maya 5, Smoothwall Corp. Guardian, What On Earth is the AFFS? RH Enterprise Linux 3

DVD HIGHLIGHTS:
Pathatlizer, Kopete,LDP, K3b , AbiWord, Scribus, CyberCafeOrganizer, Warewulf, Anteater

CDs HIGHLIGHTS:
Fedora, Vega Strike, Omnis Studio 3.1, Mandrake 9.2 updates, OpenOffice.orgQuick-starter, Rekall, The GIMP, NetPanzer

Other back issues also available, please call for further information, giving the issue number and/or date required

To order, call the back issues hotline on 0870 4448645

or +44 1458 271178 for overseas orders

Remember to quote the issue code shown above and have your credit card/switch card details ready.

Prices (inc post and packing): CD – UK£5.99, US & Europe £6.99, Rest of World £7.99
DVD – UK £6.99, US & Europe £7.99, Rest of World £8.99.

LINUX FORMAT
www.linuxformat.co.uk

Essential disc info

Read this important information before you use your *Linux Format* coverdisc – CD or DVD. We've collated some helpful info to help you get the most from these jewels of data!

FINDING THE ESSENTIALS

MISSING SOMETHING?

As many of the programs on our discs are the very latest releases, they are often built on the very latest libraries and may depend on other packages your current Linux setup does not contain. We try to provide you with as many of these important supporting files and libraries as possible, though obviously we don't have space to include absolutely everything.

In many cases, the latest libraries and other packages you might need will be included in the "essentials" folder on the disc, so if you are missing dependencies, this is the first place to look.

PACKAGE FORMATS

Wherever possible, we try to include as many different types of package for an installation as possible, whether that be distribution specific RPMs, debs or whatever. Please bear in mind that we can only do this where space permits and when the packages are available.

We will, apart from exceptional or legally restricted situations, include the source files for any package, so that you can build it yourself.

DOCUMENTATION

These pages provide helpful information on how to install and use some of the packages on the CD. Please note that many of the applications come with their own documentation, and there are additional notes and files in the relevant directories.

WHAT ARE ALL THESE FILES?

If you are new to Linux, you may find the profusion of different files and extensions confusing. As we try to give as many packages as possible for compatibility, there will often be two or three files in a directory covering different types of Linux, different architectures and usually source and binary versions – so which do you install? They can be identified by their filenames, and usually just by the file extensions.

Someap-1.0.1.i386.rpm – This is probably a binary rpm, designed to run on x86 systems.

Someap-1.0.1.i386.deb – The same, but a debian package.

Someap-1.0.1.tar.gz – This is usually source code.

Someap-1.0.1.tgz – Same as the above, tgz is abbreviated form of tar.gz

Someap-1.0.1.tar.bz2 – Same, but uses bzip2 compression instead of zip

Someap-1.0.1.src.rpm – This is also source code, but supplied as an rpm to make it easier to install

Someap-1.0.1.i386.RH7.RPM – A binary, x86 RPM designed specifically for Red Hat Linux

Someap-1.0.1.ppc.Suse7.rpm – A binary RPM designed specifically for SuSE7.x PPC Linux.

Someap-devel-1.0.1.i386.rpm – A development version.

INSTALLING FROM TARBALLS

A tar ball is a two stage archive. First the files are archived into a single file with tar and then compressed with *Gzip* or *Bzip2*. To unpack, **cd** to the directory you want to unpack it, usually your home directory and type one of the following two lines:

```
tar xzvf /mnt/cdrom/Desktop/progname/progname-2.1.0.tgz
tar xvf --bzip2 /mnt/cdrom/Desktop/progname/progname-2.1.0.tar.bz2
```

Use the first for Gzipped files, those ending in .tar.gz or .tgz, and the second for Bzipped files, ending in .tar.bz2 or .tbz2. Naturally, you change the paths to suit the location and name of the archive. and replace **/mnt/cdrom** with whatever is applicable to your system (*eg* **/cdrom**). This normally unpacks the archive into a directory of the same name, enter that directory with:

```
cd progname-2.1.0
```

To compile and install the software, type the following three commands:

```
./configure
make
su -c "make install"
```

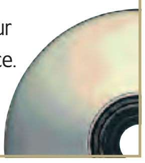
The last line will prompt you for the root password, as this stage must be run as root. If you are already logged in as root, just type **make install**. This will give you a default installation. If you want to change any aspect of the install, type **./configure —help** to see the options available. For example, you are usually able to change the default location with the PREFIX argument. When you have finished installing, you may remove the source files with:

```
cd ..
rm -fr progname-2.1.0
```

You should also log out as root, before you do anything you may later regret.

DEFECTIVE CDs

In the unlikely event of your disc being defective please email our support team (**support@futurenet.co.uk**) for further assistance. If you would prefer to talk to a member of our reader support team please call **01225 822 743**.



CREATING INSTALL CDS WITH CDRECORD

The quickest way to burn an ISO image to CD is with *cdrecord*. You need to be root to do this. First find the address of your CD-writer with

```
cdrecord -scanbus
```

This will show the devices connected to your system. The SCSI address of each device is the three numbers in the leftmost column, say 0,3,0. Now you can burn a CD with

```
cdrecord dev=0,3,0 -v
 /path/to/image.iso
```

You can simplify the command by saving some default settings in /etc/default/cdrecord. Add a line for each CD writer on your system (usually one) like this

```
Plextor= 0,3,0 12 16M
```

The first item is a label, after the SCSI address you put the speed and the buffer size to use. You can now replace the SCSI address in the command line with the label, but it gets even easier if you add

```
CDR_DEVICE=Plextor
```

Now you can burn an ISO image to disc with

```
cdrecord -v/path/to/image.iso
```

If you really don't want to use the command line, *gcombust* will do the job for you. Start it as root, select the "Burn" tab and the "ISO 9660 Image" gadget near the top of the window. Put the path to the image file in the gadget and press "Combust!". Now put on the kettle while the CD is created for you.

Other OS?

You don't have to use Linux to burn the ISO to a disc. All Linux-specific bits are already built into the image file. Programs like cdrecord simply dump it to the disk. If you don't have a CD-writer, find someone who has one, and a DVD drive, and use the CD burning software on their computer. It can be Windows, MacOS, AmigaOS whatever.

No CD burner?

What if you have no CD writer? Do you know someone else with one? You don't have to use Linux to burn the CDs, any operating system that can run a CD-writer will do the job (see above).

With some distributions it's possible to mount the images and do a network install, or even a local install from a disk partition. The methods often vary between distributions, so check vendors websites for more info. LXF

Coverdisc

Neil Bothwick is your guide through the wonders of this month's jam-packed DVD – including a program to let you watch the football when you should be working!

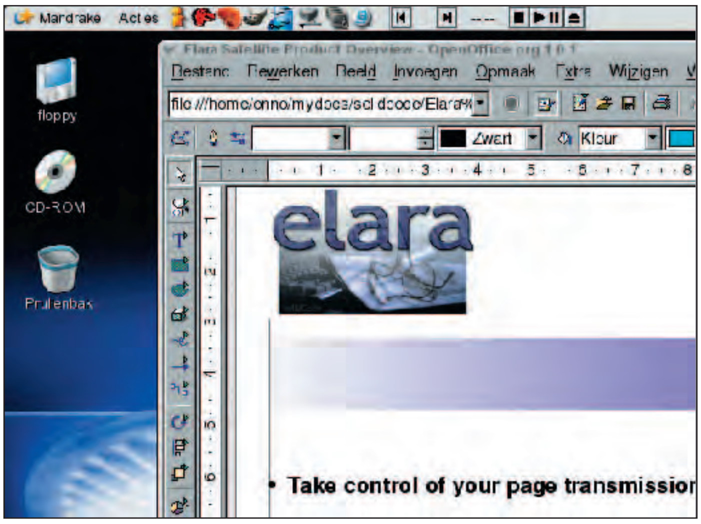
OFFICE
OPENOFFICE.ORG

We included *OpenOffice.org 1.1* on the cover of *LXF47*. This was a major upgrade from the previous version with many new features and performance improvements. With such a large project, it is inevitable that whenever you add new features, you open the door for new bugs to creep in. This is mainly a bug fix release, although it does have some optimisations too and appears to load faster. There are three archives on the DVD. Two are binary installers for X86 and PowerPC, the third is the source tarball. This is a big project that takes a long time to compile from source, well over five hours on my XP2600, so you will probably want to stick with the binary installer. CD users have no choice, there was only room for the X86 binary package.

Installation from this package is simply a matter of unpacking the archive and running the install script:

```
tar xzf /mnt/cdrom/Office/OpenOffice
.org/OOo_1.1.1_LinuxIntel_install.tar.gz
sh OOo_1.1.1_LinuxIntel_install/install
```

The **OOo_1.1.1_LinuxIntel_install** directory can be deleted once installation is complete.



Mandrake
Acties
floppy
CD-ROM
Prullenbak
Flara Satellite Product Overview - OpenOffice.org 1.0.1
Bestand Bewerken Beeld Invoegen Opmaak Extra Wijzigen
file:///home/onno/mydocs/sci dcoco/Elara
Zwart
Kleur
elara
Take control of your page transmission

***OpenOffice.org* gets another update. It is mainly bug fixes, but they seem to have slipped some speed improvements in there too.**

SYSTEM
XORG-X11 & XFREE86

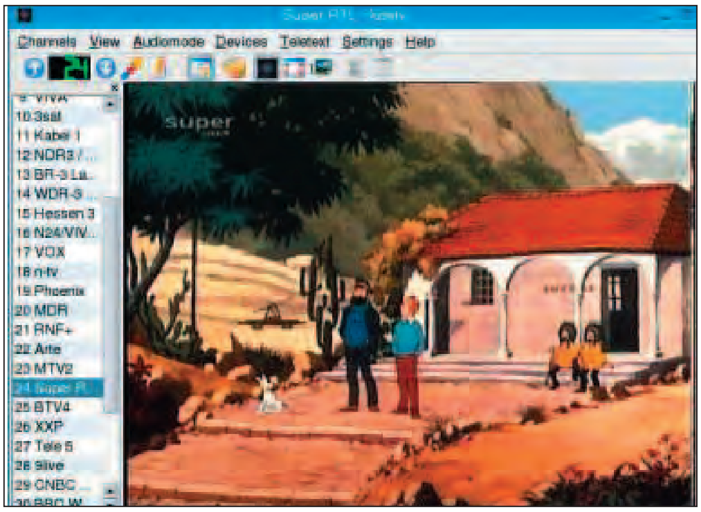
It's been hard to miss the controversy over the latest release of XFree86 and its licence. The prevailing opinion is that the new XFree86 licence is incompatible with the GPL, so most distros are not including it. Some are switching to the new release from the **X.org** Foundation. This is based on the second release candidate of XFree86 4.4, the last one to be released under the old licence. It has been updated, it's not simply a re-release of an old beta. In fact, it's proved to be very stable, I'm writing this on an X.Org desktop.

The controversial XFree86 licence does not prohibit distribution of the source and more than the old version. Its changes affect binary distributions that include linked XFree and GPLed software. This means there is no reason for us not to include the new 4.4 release here for you to try. Linux is all about choice, and now you have a choice of two new releases of X. There is not much difference yet, as both have only had a short development period since the code forked, but that will change.

Each system has its source code spread across seven tarballs. Extract them all into the same directory. This will create an *xc* directory containing all the source. Installation instructions are contained in a file called BUILD or INSTALL in this directory. Read these carefully before attempting to compile and install your system. This may be stating the obvious, but X is a major system component and an error while installing a new version could stop X working and leave you with only a text console. The safest precaution is to back up your / partition before installation.

GRAPHICS
KDETV

Will sales of TV cards increase this summer? With Euro 2004 and the Olympic Games getting plenty of screen time, how many people will take the opportunity to sneak in some extra viewing while pretending to be getting on with the household accounts or writing some letters? There are a few names used when discussing TV viewing software, and this program has had most of them.



A TV viewer by any other name – *kdetv* is now a mature and useful program, ideal for watching the footie when you're pretending to work. Just remember to keep the volume turned down!

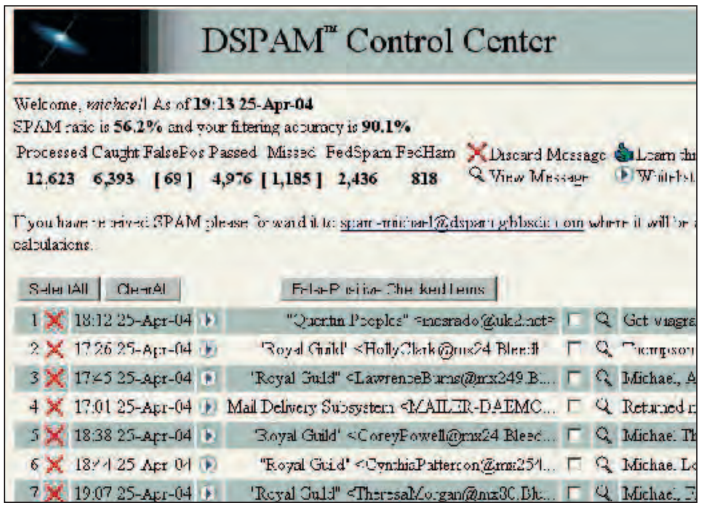
On the DVD

Wherever you see this logo it means there's related stuff on the DVD

IMPORTANT NOTICE

Before you even put the DVD in your drive, please make sure you read, understand and agree to the following:

The *Linux Format* DVD is thoroughly tested for all known viruses, and is independently certified virus-free before duplication. We recommend that you always run a reliable and up-to-date virus-checker on ANY new software. While every care is taken in the selection, testing and installation of DVD software, Future Publishing can accept no responsibility for disruption and/or loss to your data or your computer system which may occur while using this disc, the programs or the data on it. You are strongly advised to have up-to-date, verified backups of all important files. Please read individual licences for usage terms.



DSPAM™ Control Center

Welcome, *michaell* As of **19:13 25-Apr-04**
SPAM ratio is **56.2%** and your filtering accuracy is **90.1%**

Processed	Caught	FalsePos	Passed	Missed	FedSpam	FedHam
12,623	6,393	[69]	4,976	[1,185]	2,436	818

If you have received SPAM please forward it to: spam-michael@dspam.gblbsdc.com where it will be a
calculations.

1	18:12 25-Apr-04	"Quentin Peoples" <mcsrado@uk2.net>	Get viagra
2	17:26 25-Apr-04	'Royal Guild' <HollyClark@mx24 Bleed	Thompson
3	17:45 25-Apr-04	'Royal Guild' <LawrenceBurns@mx249.B...	Michael, A
4	17:01 25-Apr-04	Mail Delivery Subsystem <MAILER-DAEMO...	Returned m
5	18:38 25-Apr-04	'Royal Guild' <CoreyPowell@mx24 Bleed...	Michael. Th
6	18:44 25 Apr 04	"Royal Guild" <CynthiaPatterson@mx254...	Michael. Lo
7	19:07 25-Apr-04	'Royal Guild' <TheresaMorgan@mx30.Ble...	Michael, T

***DSPAM* provides another means of detecting spam in your mailbox – although with the rapidly increasing quantities of unsolicited bulk emails, it's probably quicker to look for non-spam now.**

Starting life as *KWinTV* before becoming *QTVision* and now called *kdetv*, this program has had more name changes than a *Mozilla* browser. Jesting aside, this program – whatever you call it – does a good job. Using a familiar KDE layout, *kdetv* presents an intuitive and easy-to-use interface, The Channel Wizard will scan for stations and either add them to an existing channel list or create a new one. The Channel Editor window makes it easy to edit the channel settings, adding suitable names for each channel and putting them in whatever order you prefer. The latest version, 0.8, is only available as a source code tarball, but installation is simple, using the standard **./configure && make && make install** incantation.

This program has no particular dependencies beyond KDE3 (and a TV card of course). Users of RPM-based systems may need to add the *-devel* RPMs if the **./configure** stage fails. The devel packages are not needed to run programs but are usually necessary when compiling from source. Your distribution's package manager should be able to take care of installing them for you.

INTERNET

DSPAM

When people were forecasting that spam would reach 50 per cent of all email volume by the early part of this year, others called them alarmist. Now those predictions have come true and the amount of spam is *still* increasing. There's not much you can do to stop spam being sent to you once these parasites have got hold of your email address, but you can filter it out on receipt. This at least avoids wasting storage space and – more importantly – your time in looking at and deleting it. It is also important if you have children, as much of it is quite unsuitable for them to receive, not to mention being offensive to most adults.

SpamAssassin has long been considered the ultimate spam killer, but it appears to have become a victim of its own success. Just as recipients are doing their best to filter spam, the spammers are trying ever harder to beat the filters. With the increased usage of *SpamAssassin*, it has become the one to beat. It seems that a lot more spam gets past it than before, even with everything kept fully up-to-date.

DSPAM (as in de-spam) takes a different approach to filtering, so is not susceptible to the workarounds for more established filters. *DSPAM* is a mail delivery agent (MDA) like *procmail*. It can be used as a replacement for *procmail*, still delivering the mail to users' mailboxes, but filtering the spam first. The default action is to move detected spam into a quarantined mailbox, that can be checked for false positives. As an alternative, you can have have it pass all mail to the user, but tag spam in the headers, in much the same way that *SpamAssassin* does.

If you only use *procmail* to deliver mail, and maybe pass it through a spam filter first, *DSPAM* can replace it. If you use – and want to keep using – some of the other features of *procmail*, *DSPAM* has an option to pass a mail back to *stdout* after checking. Using this method, you can call it from a *procmail* recipe. Much of the behaviour is set at compile time (the latest beta places less reliance on this and uses more run time configuration), so read the documentation carefully before running **./configure && make && make install**, you will probably want to add some options to **./configure**.

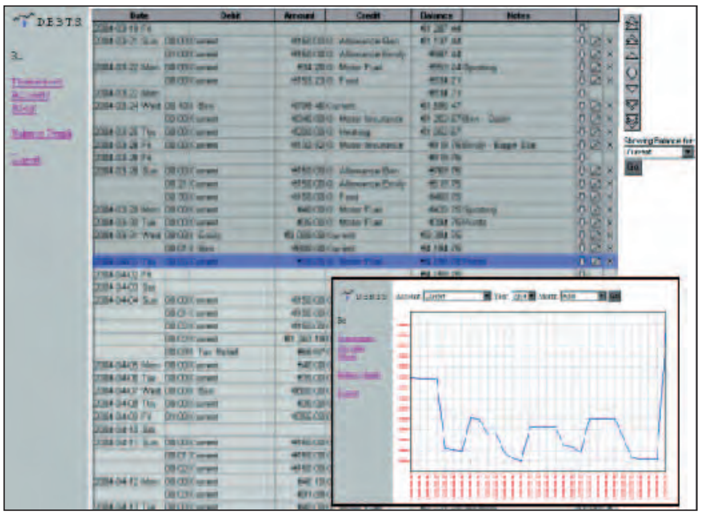
Though *DSPAM* is intended for server use, any mailer with the facility to pass a mail through a pipe while filtering should be able to use it locally.

OFFICE

DEBTS

There are two ways to increase the chances of some software on our coverdiscs also getting a mention in these pages. One is to make it so useful and attractive that we cannot resist mentioning it, the other is to give it a name that is a serious contender for the Contrived Acronym of the Month (CAM) award. Giving an accounting program a name like *Double Entry Based Transaction System*, so that its acronym becomes *DEBTS*, seem like a blatant attempt at the latter. This doesn't mean that *DEBTS* is a bad program, but it is difficult to make accounting software seem exciting and sexy, except maybe to accountants. Nonetheless, keeping track of your accounts is important, if only to ensure that you have enough funds to build your own AMD64 ultimate Linux box and still have money left over for beer and pizza.

This is a web-based system using PHP and MySQL, so you will need these – and *Apache* – installed on your system. There is one major advantage in using such a system, as opposed to the local data files that something like *GnuCash* (also on the DVD) uses. You are using a single data source, accessible over a network. So you can run the program on your desktop but access the same program and information from a laptop while watching TV. You could even make it accessible over the Internet if you have a permanent Internet connection, but you should think very carefully about doing this. You don't want to run the risk of anyone else gaining access to your accounts! LXF



A somewhat contrived name, but it's still a useful program. *DEBTS* accounting displays its tables of figures in a browser.

CREATING YOUR OWN DYNE:BOLIC CD

As noted elsewhere, if you have the Dyne:bolic disc in your DVD drive, you will not be able to change discs to play a DVD with *Xine* or *MPlayer*. So you may want to transfer it to a CD. We have provided the usual *mkiso* and *winmkiso* scripts, to create an ISO image from Linux or Windows respectively. The easiest way to use this is from within Dyne:bolic itself. Your hard disk partitions are automatically mounted under /vol, so you can write the ISO image to the root of /dev/hda1 with:

```
sh /mnt/dynebolic/Distros/
DynebolicGNU-Linux/mkiso /vol/hd1
```

DVD CONTENTS AT A GLANCE

Desktop

Alexandria	A book collection manager for GNOME
Allin1	*FluxBox* applet to monitor CPU, memory, net, filesystems
AntiRight	A lightweight desktop environment
Cdargs	Adds bookmarks and browsing to the shell command 'cd'
Cdrtools	A tool to create disk-at-once and track-at-once CDs
CommandOutputColourer	Applies matching rules to display command output in colour
CSed	A colour stream editor
EventWatcher	Notifies you about events collected by plugins
FontForge	A font editor for TrueType and PostScript fonts
GenerateBackground	A random wallpaper selector
Grdesktop	A GNOME frontend for the rdesktop client
HalogenDesktop	An elegant, lightweight desktop environment for Linux
Ion	Tiling tabbed window manager designed for keyboard users
Joe	A Free ASCII-Text Screen Editor for UNIX
Konstruct	Installs KDE with *KOffice*, *KDevelop*, and *Quanta Plus*
KPopup	Sends and receives Winpopup messages
Krusader	Twin-panel file manger for KDE 3
LogTool	Syslog file parsing, report generation and monitoring
Lpe	Small, fast console mode programming editor
MountISOImage	A KDE service menu for mounting ISO images
Pogo	Highly customizable application launcher for X11
Pydspam	A Python wrapper for *dspam*
Revelation	A password manager for GNOME 2
Saddle	An intelligent GUI for mounting/unmounting devices
Sven	Assign keys on multimedia keyboards
TkDesk	Tk/tcl-based X11 Desktop/File manager
Type1URWFontsWithCyrillics	Type1 URW fonts extended with Cyrillic glyphs
XRoar	A Dragon 32/64 and Tandy Coco emulator
XSensors	A graphical sensors display application

Development

AnjutaIDE	An Integrated Development Environment for GNOME
CherryPy	A tasty toolkit for pythonic Web development
CUPSDriverDevelopmentKit	A driver development kit with samples for *CUPS*
CutePHPLibrary	A PHP functions library
Dnspython	A DNS toolkit for Python
Gtk2-Perl	Perl bindings for GTK+ 2.x
Jikes	Translates Java source into bytecoded instruction sets
ManEdit	Man page editor, viewer and browser
MemCheckDeluxe	A memory usage tracker and leak finder
MIT-Scheme	A programming environment for *Scheme*
PyX	A Python graphics package
Ruby	An object-oriented language for quick and easy programming
Snmp-extension	A Net-SNMP extension for giving qdisc/class statistics
wxPython	A Python extension module for *wxWidgets*

Distros

Clash	A bootable CompactFlash system builder
CompactFlashLinuxProject	Distro to run on a compact flash card in read-only mode

Graphics

Arabeske	An arabesque-like pattern design tool
AV-Convert	Record, convert, and edit video files using a C++ interface
AverFunTVLiteLinuxDriver	A driver for an ISA TV tuner card
AVInfo	A utility for examining AVI and MPEG files
Coriander	A 1394 digital camera controller
Dvd2divx3pass	Yet another DVD 2 DiVX ripping program
Dvd::rip	A full featured DVD Ripper GUI
Dvtext	Renders DV movie clips of titles and scrolling credits
eLViS	Watch JPEGs captured by motion
File2divx3pass	*MEncoder* frontend to convert any media-file to DivX
Gnuplot	Plotting package with many output formats
Graphics3D	A high-performance 3D graphics codebase
GraphicsMagick	A comprehensive image processing and 2D rendering system
ImageMagick	Automated and interactive manipulation of images
ImgSvr	Web image server to browse digital images
Jpgind	A commandline JPEG Web gallery tool
KavlonColoringBook	A colouring book
MPlay	A console-based *MPlayer* front-end
Mplayerplug-in	A video plugin for *Mozilla*-based browsers
MyPHOTOS	A photo blog
OptiPNG	A lossless PNG optimizer
Spit	A graphical picture indexing tool
TheGallery	A slick Web-based photo album written using PHP
Tuxrip	A Linux bash script for ripping and encoding DVDs
Unicap	A capture library
Univision	An image capture application
Video21394ConverterDriver	A driver for The Imaging Source's video to 1394 converter
Zr364xxForLinux	A kernel module for Zoran chips based USB/JPEG cameras

Internet

ACPModemDriver	A Linux driver for the ACP (Mwave) modem
Annuaire	Allows you to manage a private directory on Internet
AscensionEmail	A KDE email client
BitTorrentQueueManager	A console-based *BitTorrent* client
DownloaderForX	Downloads files from the Internet via both FTP and HTTP
DSPAM	A server-side anti-spam agent for UNIX email servers
DSPAMphpControlCenter	A *DSPAM* quarantined message checker
GeekCredit	A secure peer-to-peer digital currency
GlobespanADSLModemDriver	A *Globespan*-based USB ADSL modems driver
Icpld	Log the performance of your Internet connection
KDEMassMailer	A powerful Mass Mailer for KDE 3.x
MozillaThunderbird	A total redesign of the *Mozilla* mail component
Nameko	A one-file Web mail script
NATMonitor	Monitors local hosts' Internet bandwidth usage
NatRail	Query UK train times via the command-line or a Web page
Walker	Match your remote website to the local copy
Wput	The opposite of wget
YahooGroupArchiver	A command line interface to interact with Yahoo! Groups

Mobile

Bbacpi	A tool for X11 that shows laptop battery information
Ifplugd	An ethernet link beat detection daemon

Office

GnuCash	A program to keep track of your finances
Kexi	KDE app to potentially act as a replacement for Access
MailManager	Software which helps teams deal with large volumes of email
OOoReport	A report writing tool for *OpenOffice.org*

Server

Alecto	Template-based Content Management System
BASTED	A spam honeypot
Bluefish	A GTK-based Web development editor
Exim	A Message Transfer Agent for Unix systems
Exim-Python	An *Exim* extension for executing Python functions
Geximon	A *GTK2* monitor for the *Exim* MTA
HostingBackup	A set of scripts for backing up your hosted Web sites
MIMEDefang	Anti-virus mail filter
Moftpd	A powerful FTP server
MySQLBackupPro	Creates and restores backups of *MySQL* databases
NullWebmail	A simple POP3/SMTP webmail CGI
OneTimeDownload	A script to allow downloading files using a one time ticket
phpMyAdmin	Handles the basic adminstration of *MySQL* over the WWW
PyEximon	A GNOME monitor and manager for *Exim*
Renattach	Renames or deletes dangerous email attachments
SWISH++	File indexing and searching engine
ThomersMusicVault	A Linux streaming music (MP3/Ogg/FLAC) server
Vsftpd	A very secure and fast GPL'd FTP server

Sound

Abcde	A better CD encoder
Ags	An audio composing application
BURN	A quick command line tool for writing CDs
burnCDDA	A tool for creating audio CDs
Cddb2md	Fetches track title information and writes it to a MiniDisc
Drums++	A programming language for creating MIDI drum sequences
Gnomoradio	A peer-to-peer music player for GNOME
Grip	A CD player and CD ripper/encoder
Jinzora	A Web-based MP3 jukebox
M3Tk	A tool that generates M3U files and edits ID3 tags
MP3c	An Audio-CD to MP3/OGG-Converter, with use of *CDDB*
MusicalMIDIAccompaniment	A MIDI track accompaniment generator
MusicManager	A *Konqueror* plugin for management of audio files
Musicus	A command-line music player utilizing xmms plug-ins
NomadJukeBoxKIO::Slave	An I/O plugin for using your *Nomad Jukebox* in KDE
OsdFoo	An external plugin for *XMMS*
P2P-Radio	A peer-to-peer audio and video broadcasting system
PhilipsMusicWriter	A text-to-PostScript music typesetter
Qjackctl	*Qt* application to control the *JACK* sound server daemon
SimpleMultitrack	A multitrack audio recorder
Streamtuner	A *GTK+ 2.0* stream directory browser
WavTools	A set of simple WAV processing tools
XMMSDVBInputPlugin	Enables *XMMS* to directly play and record DVB audio services

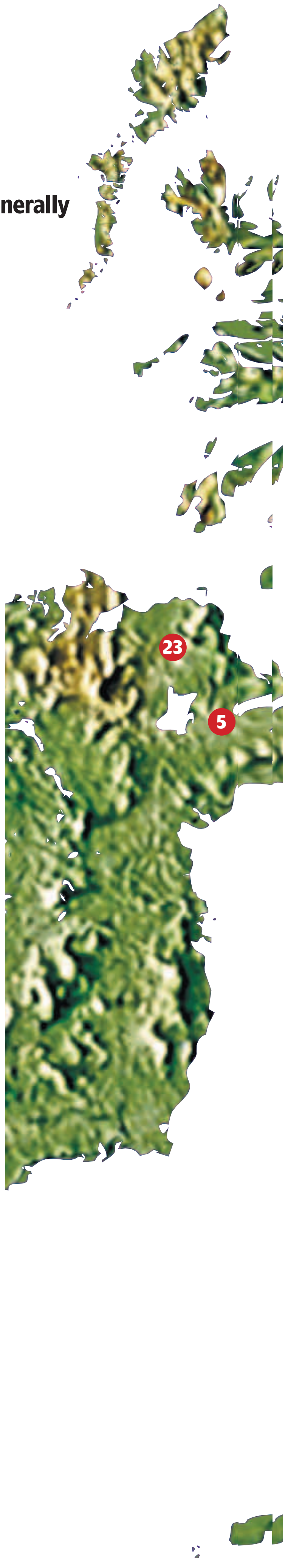
System

AddUser-NG	Adduser script with more functionality, such as plugins
Apt-got	Builds and supervises a partial (or full) Debian mirror
CDbackup	A CD-R(W) backup utility
Chkrootkit	Locally checks for signs of a rootkit
ClamAntiVirus	An anti-virus utility for Unix
Deborphan	A Debian orphaned library finder
E2undel	An undelete tool for the ext2 file system under Linux
Emerde	A port of Gentoo's Portage system for other distributions
FSHeal	A tool for mending filesystems and reporting statistics
HydraBackupSystem	A Python backup system
Idlebeep	Beeps if your system goes idle, surprisingly!
Monit	A utility for monitoring Unix system services
RootkitHunter	A file scanner for rootkits, backdoors, and sniffers
SynapticsTouchPadDriver	A Synaptics Touchpad driver for XFree86

PLUS 9 GAMES!

User Groups

LUGs worldwide are full of members keen to help with your problems, discuss ideas, and generally chat about all things Linux. You can find lots more information online at www.lug.org.uk



1 ABERDEEN
www.aberdeen.lug.org.uk/

2 ANGLIAN LUG
www.alug.org.uk/

3 AYLESBURY
drbond@educational-computing.co.uk/

4 BEDFORDSHIRE
www.beds.lug.org.uk/

5 BELFAST & N. IRELAND
bgalbraith@gmx.net/

6 BERKSHIRE & THAMES VALLEY
www.sclug.org.uk/

7 BEVERLEY
www.beverley.lug.org.uk/

8 BIRMINGHAM (SOUTH)
www.sb.lug.org.uk/

9 BLACKBURN
matt@consultmatt.co.uk/

10 BLACKPOOL
www.blackpool.lug.org.uk/

11 BORDERS (ENGLAND/SCOTLAND)
welby@mcrobers.com.kg

12 BRIGHTON
www.brighton.lug.org.uk/

13 BRISTOL & BATH
www.bristol.lug.org.uk/

14 BRIXTON
blug@communitytechnology.org.uk/

15 CAMBRIDGE
mark@taurine.demon.co.uk/

16 CHESHIRE (SOUTH)
www.sc.lug.org.uk/

17 CHESTERFIELD & DISTRICT
www.spirelug.org.uk/

18 CHICHESTER HARBOUR – MARITIME GEEKS
hugh@mjr.org

19 CLEVELAND OPEN SOURCE GROUP
www.cleveland.lug.org.uk/

20 COVENTRY
www.covlug.org.uk/

21 CUMBRIA
www.cumbria.lug.org.uk/

22 DERBYSHIRE (SOUTH)
www.sderby.lug.org.uk/

23 DERRY
www.derry.lug.org.uk/

24 DEVON & CORNWALL
www.dclug.org.uk/

25 DONCASTER & SCUNTHORPE
www.doncaster.lug.org.uk/

26 DORSET
www.dorset.lug.org.uk/

27 DUNDEE & TAYSIDE
www.dundee.lug.org.uk/

28 EDINBURGH
www.edlug.org.uk/

29 EXETER (UNIVERSITY)
nick@urgusabic.net/

30 FALKIRK
www.falkirk.lug.org.uk/

31 GLASTONBURY
www.glastonbury.lug.org.uk/

32 GLOUCESTERSHIRE & COTSWOLDS
www.gloucs.lug.org.uk/

33 HAMPSHIRE
www.hantslug.org.uk/

34 HEREFORD
www.hereford.lug.org.uk/

35 HERTFORDSHIRE
www.herts.lug.org.uk/

36 HERTFORDSHIRE (EAST)
www.ear.lug.org.uk/

37 HUDDERSFIELD
www.hud.lug.org.uk/

38 HULL
www.hull.lug.org.uk/

39 ISLE OF MAN
www.iom.lug.org.uk/

40 ISLE OF WIGHT
www.iow.lug.org.uk/

41 KENT
www.kent.lug.org.uk/

42 LANCASTER
www.lancasterlug.org.uk/

43 LEAMINGTON SPA
ian@siim.info

44 LEICESTER
www.leicester.lug.org.uk/

45 LINCOLNSHIRE
www.lincs.lug.org.uk/

46 LONDON (GREATER - GLLUG)
www.gllug.org.uk/

47 LONDON (LONIX)
www.lonix.org.uk/

48 LONDON (NORTH-WEST)
www.northwestlondon.lug.org.uk/

49 LONDON (SOUTH)
www.sl.lug.org.uk/

50 LOUGHBOROUGH
www.loughborough.lug.org.uk/

51 LUTON & ST ALBANS
www.lust.lug.org.uk/

52 MALVERN
www.malvern.lug.org.uk/

53 MANCHESTER
www.manlug.mcc.ac.uk/

54 MANSFIELD
www.mansfield.lug.org.uk/

55 MENAI
kevin@dotmon.com

56 MILTON KEYNES
www.mk.lug.org.uk/

57 MORAY
Stu@forres.co.uk

58 NORTH EAST ENGLAND (DURHAM)
www.ne.lug.org.uk/

59 NORTH LONDON
jason@voyagercomputers.co.uk

60 NORTHANTS
www.northants.lug.org.uk/

61 NOTTINGHAM
www.nottingham.lug.org.uk/

62 NUNEATON
www.nuneaton.lug.org.uk/

63 ORMSKIRK
www.ormskirk.lug.org.uk/

64 ORPINGTON
www.orpington.lug.org.uk/

65 OXFORD
www.oxford.lug.org.uk/

66 OXFORD BROOKES UNIVERSITY
www.obaoss.lug.org.uk/

67 PETERBOROUGH
www.peterboro.lug.org.uk/

68 PRESTON & LANCASHIRE
www.preston.lug.org.uk/

69 RUGBY
www.rugby.lug.org.uk/

70 SALFORD (UNIVERSITY)
http://linsoc.ussu.salford.ac.uk/

71 SCARBOROUGH
www.scarborough.lug.org.uk/

72 SCOTTISH
www.scottish.lug.org.uk/

73 SHEFFIELD
www.sheflug.org.uk/

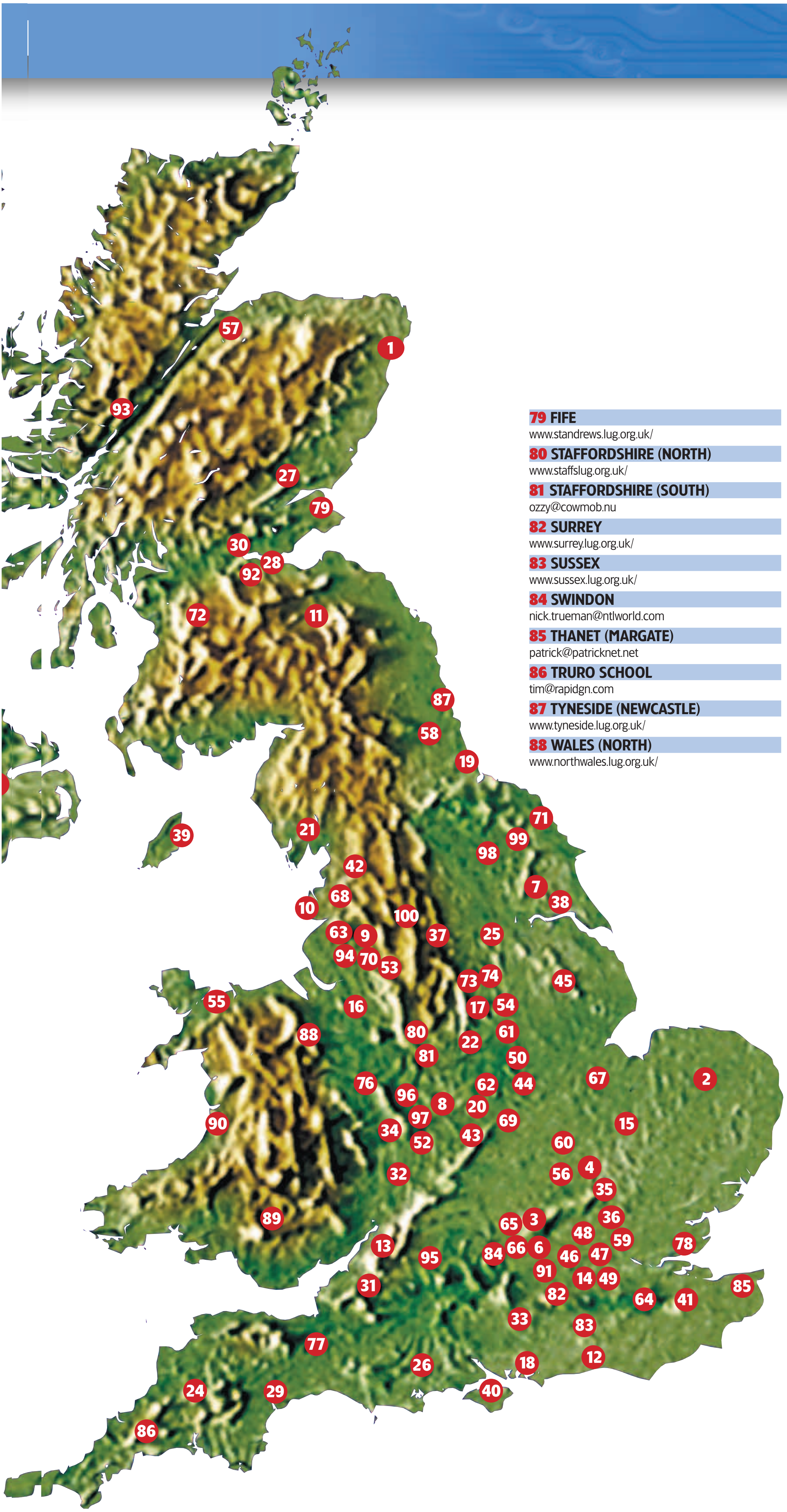
74 SHEFFIELD LINUX & BSD USER GROUP
www.slugbug.org.uk/

75 SHETLAND
www.shetland.lug.org.uk/

76 SHROPSHIRE
www.shropshire.lug.org.uk/

77 SOUTH WEST
www.southwest.lug.org.uk/

78 SOUTHEND-ON-SEA
derek@techsys.co.uk



79 FIFE
www.standrews.lug.org.uk/

80 STAFFORDSHIRE (NORTH)
www.staffslug.org.uk/

81 STAFFORDSHIRE (SOUTH)
ozzy@cowmob.nu

82 SURREY
www.surrey.lug.org.uk/

83 SUSSEX
www.sussex.lug.org.uk/

84 SWINDON
nick.trueman@ntlworld.com

85 THANET (MARGATE)
patrick@patricknet.net

86 TRURO SCHOOL
tim@rapidgn.com

87 TYNESIDE (NEWCASTLE)
www.tyneside.lug.org.uk/

88 WALES (NORTH)
www.northwales.lug.org.uk/

89 WALES (SOUTH)
www.swlug.org.uk/

90 WALES (WEST)
www.westwales.lug.org.uk/

91 WALTON ON THAMES
rael@freeuk.com

92 WEST LOTHIAN
ben@150bpm.co.uk

93 WEST OF SCOTLAND
woslug@uk2.net

94 WIGAN AND ST. HELENS
www.all-the-johnsons.co.uk/wishlug/

95 WILTSHIRE
www.wiltshire.lug.org.uk/

96 WOLVES
www.wolveslug.org.uk/

97 WORCESTERSHIRE
www.worcs.lug.org.uk/

98 YORK
www.york.lug.org.uk/

99 EAST YORKSHIRE
sharkonline@whatemail.com

100 WEST YORKSHIRE LUG
www.wylug.org.uk/

OTHERS

BENELUX
www.b-lux.org/

BUSINESS LUG
www.businesslug.org/

CHANNEL ISLANDS
myles.hartley@db.com

FUENGIROLA GNU/LUG
www.fuengirola.lug.org.uk/

SCHOOLS
www.schools.lug.org.uk/

STUDENTS
www.students.lug.org.uk/

TR LINUX
www.trlinux.lug.org.uk/

UK UNIX USER GROUP (UKUUG)
www.ukuug.org/

YOUNG LINUX USER GROUP
www.young.lug.org.uk/

ORG OF THE MONTH

Want to start your own LUG? www.lug.org.uk/ can help!

The website was originally the idea of Mark Lewis, created in 1997 when he registered the **lug.org.uk** domain name and offered free registrations to user groups. The site has 2 primary objectives:

1 To introduce users of Linux to others via the concept of Linux User Groups.

2 To freely allow Lugmasters to set up their own local LUG with our aid using the **lug.org.uk** domain name.

Now hosting around 100 LUGs, it is **lug.org.uk**'s ultimate goal to have one User Group per county in the UK. The site has become a core resource for LUG co-ordinators, and has grown so much that it is now managed by a team of volunteers, with Mark at the head.

In an effort to help offer new and existing Lugmasters the opportunity to promote Linux, **lug.org.uk** is happy to provide services for FREE – the only 'strings' being that your site and services must *only* be used to set up, manage and maintain a User Group for other local enthusiasts. **lug.org.uk** is happy to help with services such as:

- Subdomains in the format . **groupname.lug.org.uk**
- A login to the **lug.org.uk** server for email and web hosting
- DNS to forward A and MX records to your own mail and web servers
- POP3 mail and websites **groupname@lug.org.uk** and **www.groupname.lug.org.uk**
- Email aliases such as **webmaster@groupname.lug.org.uk**
- Mailman Mailing lists for user discussion and debate
- *MySQL* & PHP database services
- Invitation to join the Lugmasters mailing list for help and support

Should you like to ask any questions about how to start a Linux User Group in your area – whether you are in the UK or not – fire off an email via the website's contact page and **lug.org.uk** will be happy to oblige. The dedicated and hard-working team helps the Linux User Group organisers to run their LUGs, so that they can in turn help all of us get the most out of our Linux experience.

LUG pages – use 'em or lose 'em

Linux Format needs the LUGgers of the world – and the Linux User Groups need *Linux Format*! Whether you're a member of a LUG, want to join your local one or set your own up, we want to hear from you. Jono Bacon explains the situation...

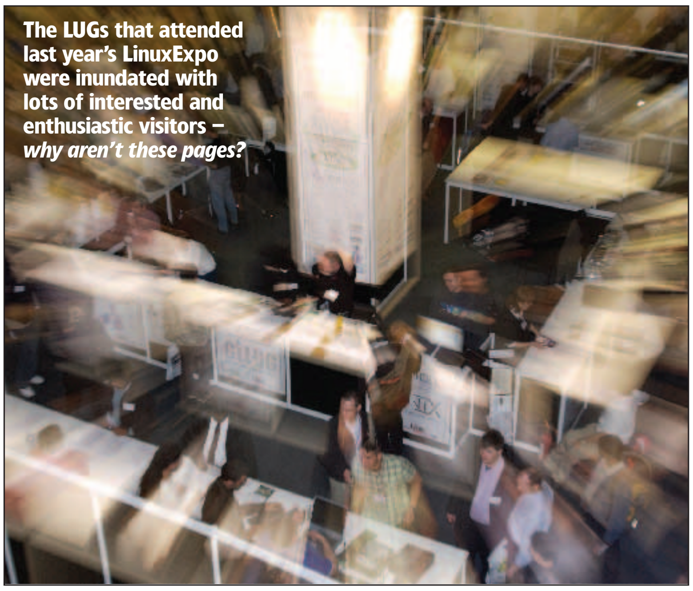
Regular readers of *Linux Format* will be well aware of these few pages wedged at the back of the magazine dedicated to LUGs. These pages are not only looking to help support and inform readers on what LUGs are available, but to also focus on LUGs and how they work.

It is an unfortunate fact that over the last six or so issues, these three pages have been generating very little response and feedback from LUGs and we need to change that! We are looking to really make these few pages a truly useful and insightful foray into LUG culture – both in the UK and internationally – but without any coherent input to work from, we are finding it more and more difficult to justify the space in the magazine. It would be a shame to see the LUG pages disappear into the ether, so we are appealing for your help.

You may be wondering what kind of feedback would be useful. We are ideally looking for events, seminars, special meetings, reports, stories, announcements and other kinds of LUG-related news. The key thing is in making these pages a great outlet to get new people along to your LUG. In a similar way, if you are new to the idea of LUGs, these pages should help you get involved more easily.

New to LUGging?

For those who have never ventured along to a LUG meeting, it can be a rather nerve-racking proposition at first. Before my first meeting, I had visions of unwieldy beards, sandals, and conversations that would sail over my head. I was delighted to find that the LUG (Northants LUG back then) were very friendly and very welcoming. Most of the meeting was spent talking and debating about the software we use every day, along with a lot of generally interesting chit-chat! From my experience of visiting LUGs and actually running my own LUG (Wolverhampton LUG), I have only ever met a few people that have not enjoyed going along, preferring to reside only as a name on their LUG mailing list. LUGs are generally filled with people who love to tinker with Linux as much as you do, but be social with it – and chances are, they'll share many of your other interests too.



The LUGs that attended last year's LinuxExpo were inundated with lots of interested and enthusiastic visitors – *why aren't these pages?*

One of the main concerns a lot of new users to Linux seem to have is a nervousness that they will look a little silly if they don't know a lot about Linux in front of more knowledgeable people. There really is no reason to feel uncomfortable about going to a LUG – we all start somewhere, and learning is the LUG's *raison d'etre*, after all! You'll probably find that you pick up more info in the space of a few hours with a LUG than you would poring over the Internet for days on end.

LUGs are great social, technical and fun groups to be a part of, and we're sure you will have a great time. Keep *LXF* posted with how you get on, and existing LUG-goers – send us (with at least 5 weeks notice, please!) your event info, appeals, announcements *etc*. *LXF* wants to print MORE of these LUG pages, not none at all! LXF

Linux User Group information

Please send your LUG-related ideas, event details, criticisms, comments, wants, needs *etc* to: **LUGS! *Linux Format*, 30 Monmouth Street, Bath, BA1 2BW** or email your stuff to: **lxf.lugs@futurenet.co.uk**

EDITORIAL
Editor Nick Veitch *nick.veitch@futurenet.co.uk*
Reviews Editor Paul Hudson *paul.hudson@futurenet.co.uk*
Art Editor Julian Jefferson *julian.jefferson@futurenet.co.uk*
Production Editor Matt Nailon *matthew.nailon@futurenet.co.uk*

Editorial Contributors Jono Bacon, Andy Channelle, David Cartwright, Michael J Hammel, Mike Saunders, Hoyt Duff, Jon Kent, Richard Drummond, Marco Fioretti, Andy Hudson, David Coulson, Pamela Jones

COVER CD PRODUCTION
CD Editor Neil Bothwick

ART CONTRIBUTORS
Art assistance Colin Nightingale
Photography Powerstock, Neil Godwin, Photodisc, Eyewire
Illustration Paul Bateman, Chris Winn, Shane Collinge

ADVERTISING SALES
Senior Sales Executive
Andrew Hudson *andrew.hudson@futurenet.co.uk*
(01225) 442244 Ext 2441

MARKETING AND PROMOTIONS
Marketing Manager Fiona Tully

PRODUCTION
Production Co-ordinator Diane Ross
Group Production Manager Clare Tovey
Commercial Buyer Mike Thorne
Advertising Production Co-ordinator Jo Crosby

MANAGEMENT
Publisher Kelley Corten
Publishing Director John Weir
Managing Director Robert Price

DISTRIBUTION AND CIRCULATION
Trade MarketingManager Jason Comber
jason.comber@futurenet.co.uk

Distributed by Marketforce (UK) ltd, 5th Floor, Low Rise Building, Kings Reach Tower, Stamford Street, London, SE1 9LS **Tel** 020 7633 3300
Overseas Distribution by Future Publishing Ltd.
Tel 01225 442244.

Overseas Licences
International Licensing Director Simon Wear
simon.wear@futurenet.co.uk **Tel** +44(0)1225 822798
Fax +44 (0)1225 788105 **Cell phone** +44(0)777 5641493

Contact Details
Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel +44 (0)1225 442244 **Email** *linuxformat@futurenet.co.uk*

Subscriptions and Mail Order
Phone +44 (0)1458 271178. See page 98
Email *linuxformat.subs@futurenet.co.uk*

All CD-ROM demos and reader submissions are supplied to us on the assumption they can be incorporated into a future covermounted CD-ROM, unless expressly stated to the contrary.

Disclaimer All tips in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your computer, peripherals or software through the use of any tips or advice.

Printed in the UK by Midway Clark (Holt) and Southern Print

Future Publishing Ltd. is part of The Future Network plc.

The Future Network produces carefully targeted specialist magazines for people who share a passion.
We aim to satisfy that passion by creating titles offering value for money, reliable information, smart buying advice and which are a pleasure to read. Today we publish more than 90 magazines in the UK, US, France and Italy. Over 80 international editions of our magazines are also published in 28 other countries across the world.
The Future Network plc is a public company quoted on the London Stock Exchange (symbol: FNET).
Non-executive Chairman Roger Parry
Chief Executive Greg Ingham
Group Finance Director John Bowman
Tel +44 (0)1225 442244

BATH • LONDON • MILAN •
NEW YORK • PARIS •
SAN FRANCISCO • WROCLAW
www.thefuturenetwork.plc.uk

future publishing
Media with passion

Next month

ISSUE 56 ON SALE WEDNESDAY 7 JULY

FASTER FILESYSTEMS

You may know a bit about ext3, and perhaps even ReiserFS – but what of the myriad other filesystems that the Linux kernel is capable of running – should you be using something else? Our comprehensive feature will show you which filesystem is best for which purpose, and open your eyes to the options available.

Which mailserver?

The best MTAs get put through their paces to discover which has the best features, which is easiest to manage and which you should trust to deliver your mail.

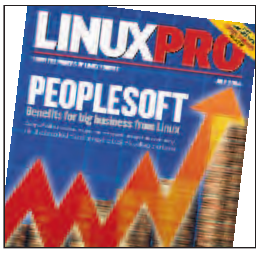
FEDORA CORE 2!

Get your hands on the new stable release of Fedora! With the 2.6 kernel, SELinux security and bags more features

PLUS

DON'T MISS NEXT MONTH'S *LINUX PRO*!

Dedicated supplement with pages of real-world Linux advice and case studies for IT professionals. Subjects covered include: storage, Red Hat Enterprise Desktop, patents, security and more!



DON'T MISS YOUR COPY OF *LINUX FORMAT*!
Get it delivered to your door every month – subscribe on page 98

The exact contents of future issues are subject to change

LINUXPRO

FROM THE MAKERS OF LINUX FORMAT

JULY 2004

PEOPLESOFT

Benefits for big business from Linux

PeopleSoft's massive migration program brings the company into the Linux fold – and brings the best HR software to Linux

PLUS

DB2 and MySQL: databases that mean business
The future of data management is here

Patent insanity for the EU
Software patents haven't been forgotten – yet!

Infosec 2004 – show report
News and products from Europe's biggest security event

Iomega back to the big time?
Will the new 35GB drive help Iomega up the storage heap?

PRACTICAL LINUX SOLUTIONS FOR I.T. PROFESSIONALS

Welcome

TWENTY PAGES OF REAL-WORLD LINUX FOR IT PROFESSIONALS

Arriving at the Infosec 2004 event early on the first day, I was one of the first to read and hear the details of the latest Department of Trade and Industry (DTI) survey on information security. There were a number of surprises (the detail is covered in more depth in our Infosec show report), mostly unpleasant ones. It seems that while companies are keen to make use of the latest technologies to give them an edge, they aren't so conscientious to make sure that the associated vulnerabilities are adequately guarded against.

One point that the survey makes is that very few people working in the IT sector have any sort of formal security qualification. That is perhaps less surprising than you think – how many security qualifications can you think of? The fact is that business is only slowly waking up to the fact that poorly managed security might not end up just costing them a bit of downtime, but could easily put them out of business. Everyone seems to be more clued up about virus risks, but very few have bothered to secure wireless networks for instance (in spite of every computing magazine that you could think of repeatedly warning of the dangers). A widely recognised security qualification might go some way to proving that security is something IT managers take seriously. Fortunately, the LPI program is working on a security-specific exam. While in the Linux world we may think we are more inherently secure and more clued up about security, it'd be nice to be able to prove it!

Enjoy this issue of *Linux Pro* – but, as ever, if there is something that you'd like to comment on, or you want to make a suggestion, just email me at the address below.

Nick Veitch Editor
nick.veitch@futurenet.co.uk

A WIDELY RECOGNISED SECURITY QUALIFICATION MIGHT GO SOME WAY TO PROVING THAT SECURITY IS SOMETHING WE TAKE SERIOUSLY

CONTENTS

4

10

CONTACTS

EDITORIAL

Editor
Nick Veitch *nick.veitch@futurenet.co.uk*
Reviews editor
Paul Hudson *paul.hudson@futurenet.co.uk*
Art editor
Julian Jefferson *julian.jefferson@futurenet.co.uk*
Production editor
Matt Nailon *matthew.nailon@futurenet.co.uk*
Publisher
Kelley Corten
Publishing Director
John Weir

Photography
Powerstock, Corbis, Digital Vision

ADVERTISING

Senior Sales Executive
Andrew Hudson
andrew.hudson@futurenet.co.uk
(01225) 442244 Ext 2441

LinuxPro is a part of
LINUX FORMAT
magazine

Published by
future publishing
Media with passion

30 Monmouth Street, Bath BA1 2BW
Tel +44(0) 1225 442244

PATENT INSANITY

WHY EUROPEAN SOFTWARE DEVELOPERS ARE TREADING CAREFULLY

As the insidious threat of software patents looms over Europe, **PAUL HUDSON** explains what all the fuss is about and why ***you*** need to make a stand...

Not since the infamous Windows worm, 'Code Red', have this many websites gone down. Several times now, web users have gone to their favourite free software page – whether that be *OpenOffice.org, The GIMP,* the FSF, Knoppix, or others – only to see the entire site front page dedicated to a protest. The battle-cry? *"No Software Patents in Europe: 10 days of web strike vs 20 years of patent nightmare!"*

The lines are drawn. On one side stands large companies with extensive arsenals of patents who stand to benefit greatly from the legitimisation of software patents, and on the other side stands the Free software community and small business where software patents threaten to crush innovation. Part of the problem that stops more people voicing their opinions is that many do not fully understand the issues at stake: patent law is often confused with copyright law, and 'intellectual property' is bandied about to the point where its meaning is almost gone. So, before we consider whether software patents are right for Europe, it's important first to clarify what patents actually are.

PATENT AND COPYRIGHT

Most would consider the term 'intellectual property' to constitute patents, copyrights, and trademarks, as well as other areas of law. As a result, just mentioning 'intellectual property' does little other than muddy the legal water surrounding these very separate issues.

Copyright covers a physical expression of your work, and automatically covers writing, music, programming code, and various other creations. Patents, on the other hand, cover ideas and thinking, and are not automatic – you need to apply for them, and the 'patentese' language that patents are written in is traditionally clouded, vague, and open to interpretation. This makes sense from the view of the patent owner, as they'd rather have their patent be applicable to more things than they had intended.

Copyright also lasts for much longer than patents – usually over 100 years, compared to 20 years for patents. Having said that, the relatively short lifespan of patents don't make them less dangerous: even five years is a very long time in computer science.

As copyright protects things that have been created, it is easily defined and abuses are fairly obvious – you would know you were breaking copyright law, for example, if you re-typed all of the *Lord of the Rings* and said you wrote it. Patents are much more difficult. If Tolkien had patented the idea of a plot-line in which a great war between good and evil that only a special hero could win, pretty much every fantasy novel would be a violation.

"SOFTWARE PATENTS DESTROY COMPETITION, LOCK IN CUSTOMERS, THROTTLE INNOVATION, AND SILENCE INDEPENDENT EXPRESSION. THE PEOPLE PUSHING FOR SOFTWARE PATENTS ARE THE LARGEST COMPANIES AND THEIR CORPORATE LAWYERS; BUT FOR SOFTWARE USERS – AND FOR ANYONE WHO CAN'T AFFORD A £2M LAWSUIT WITHOUT BLINKING – SOFTWARE PATENTS ARE ENTIRELY BAD NEWS..."

JAMES HEALD, UK CO-ORDINATOR, FFII

If it were as clear-cut as "know what patents there are, and just avoid them," things wouldn't be quite so bad. But how do you know what patents you might violate? Search a big database for things your program does? Consider US patent number 05443036, of which part of the text describes it as *"a method of inducing aerobic exercise in an unrestrained cat"*. What the patent actually covers is the process of shining a torch or laser beam against a wall so that a cat chases it. Ignoring the fact that the patent is neither original nor innovative, how would you have located it if you were thinking of making your own 'cat torch'? The answer is that you couldn't, largely thanks to the obscure way in which the 'invention' is described.

SOFTWARE PATENTABILITY

Consider what would happen if ideas in software could be patented. At first, it might seem like a smart idea – think of all the poor, starving inventors who have lived on naught for years while working on new software ideas. However, what would have happened if the first company to create a word processor with word-counting functionality patented it? Or if XML was covered by patents? You might scoff at the latter, but consider it was only last year when the World-Wide Web Consortium (W3C) were discussing the idea of allowing patented ideas into standards. If that proposal had gone through, software that implemented patented standards would have had to pay royalties to the patent owner – hardly something conducive to Free software.

Those who want software patents are usually large companies that have a lot to gain. The image of the poor, starving inventors is dashed by the fact that the vast majority of patents are issued to large companies. Last year alone, IBM applied for 3415 different patents, outstripping their nearest patenting competition almost 2:1; Canon filed 1,992, HP filed 1,759, and Intel 1,592. Poor? Starving? Forget it.

At the other end of the scale, smaller companies either don't have the legal resources to patent products or simply don't invent enough to warrant a patent application. As a result, the vast majority of companies are consumers of patents as opposed to producers, which places them at a competitive disadvantage to larger companies. For example, both Microsoft and IBM have a large arsenal of patents covering their ideas, but in order that they may both avoid legal trouble they cross-license with each other – they agree to let each other use their patents freely.

A smaller company without such an arsenal is unable to benefit from these cross-licensing agreements, effectively leaving them open to large-scale abuse by larger companies. If they dare to make a product that competes effectively with another product owned by a larger company, they are likely to find themselves being hit with lawsuits. This isn't just restricted to very large companies against very small companies – US patent 5,546,528, *"Method of displaying multiple sets of information in the same area of a computer screen"*, was filed by Adobe back in 1994, and covers the use of tabbed GUI elements much like many Linux applications have. This patent was the basis of a lawsuit between Adobe and Macromedia, which won Adobe just shy of US$3million.

Even Microsoft has been hit by patent lawsuits in its time, despite having an army of lawyers behind it whose job it is to ensure this doesn't happen. Just last year MS was told to pay $62,000,000 to Imagexpo LLC for using whiteboards inside the MS collaboration app, *NetMeeting* – something that Imagexpo had 'invented'. If even the largest software companies routinely fail to spot all the software patents out there, what hope do smaller companies have?

EU PARLIAMENTARY DISCUSSIONS

As software patents are legal in the US, we've seen such horrors as Amazon's infamous 'one-click' patent, but there have been much worse. In 2001, a patent was granted for an *"Automatic Business and Financial Transaction Processing System"* – ostensibly patenting e-commerce several years after it was first used, but written in such a generic way that even ATM manufacturers need worry. To stop these sorts of patents worming their way into Europe, the EU government has devoted substantial time to discussing the matter.

The original directive was produced by the office of Commissioner Frits Bolkestein, and was criticised by small companies, software developers, and the Open Source community for being so vague that virtually anything could be patented. The European Parliament subsequently voted to adopt several key amendments to the directive that substantially altered its meaning, including the infamous Article 6a – an amendment designed to prevent patents interfering with interoperability, which is something close to the hearts of Open Source developers.

These amendments, despite being democratically voted, were criticised by pro-patent supporters, with one saying: *"the amended proposals arguably demonstrate that the issues being debated here are too complex to be left to the European Parliament, which can hardly be expected to have in-depth expertise in patent matters"* – essentially stating that a specially selected group of patent lawyers should be allowed to decide whether to allow software patents.

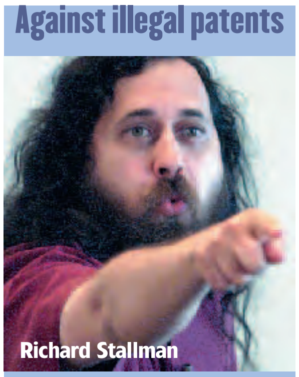
Numerous Open Source community groups and advocates have done their best to argue against software patents in Europe. Linus Torvalds and Alan Cox – two of the leading Linux developers and both European – wrote an open letter to the EU saying, *"US experience shows that, unlike traditional patents, software patents do not encourage innovation and R&D..."* and *"In particular they hurt small and medium-sized enterprises and generally newcomers in the market. They will just weaken the market and increase spending on patents and litigation, at the expense of technological innovation and research..."*

It's no surprise the Open Source community is taking such a hard-line stance against software patents, particularly as Bill Gates recently said: *"There's no question that in cloning activities, IP from many, many companies, including Microsoft, is being used in Open Source software"* – a stark warning of where unlimited software patentability could put us.

The Foundation for a Free Information Infrastructure (FFII), was behind the "No Software Patents in Europe" website protest; and to date it has organised two protest rallies where hundreds of people have turned up to campaign publicly against software patentability. In order to fight successfully against the cash-filled coffers of the pro-patent lobbyists, the FFII needs *your* help – pay a visit to its website at **http://swpat.ffii.org**. ■■■

FIGHTING THE WAR

Against illegal patents

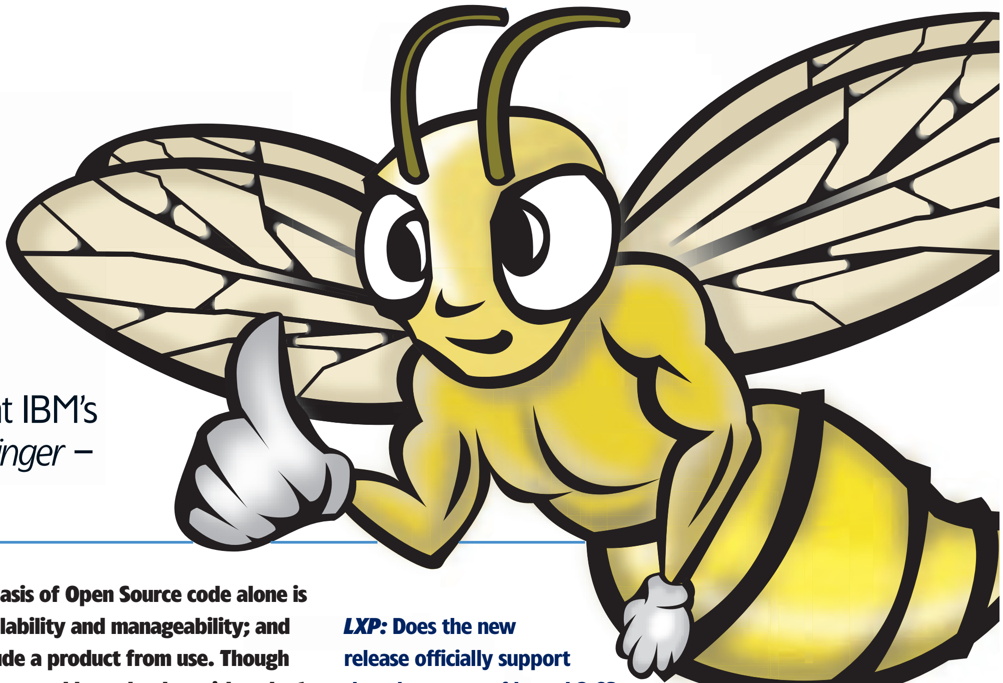

Richard Stallman

RIGHT NOW, SOFTWARE patents are illegal in Europe – whether that changes will largely depend on people in the know fighting companies who want to 'harmonise' Europe's patenting laws with America's. Sadly, in the meantime, the European Patent Office continues to allow software methods to be patented – over 30,000 patents have been granted, despite them being illegal.

Despite the pro-patent group losing two major battles (Nov 2000 and Sept 2003), they still dismiss the EU parliament's decision to keep software unpatentable as "ignorant" or "unworkable", and instead want the decision taken away from the EU and handed to the national patent offices where it will be decided in secret and by unelected officials who stand to benefit from increased patentability.

If independent and Free software development is to continue in Europe without threat from patent-laden blue-chip companies, software patents *must* be stopped. Richard Stallman summarised the current situation in a talk to the University of Cambridge Computer Labs: *"We need to make management aware of what software patents will do to them. Get their support in fighting against software patents in Europe. The battle is not over. It still can be won."*

IT'S A STING



PAUL HUDSON takes a look at what IBM's new release of *DB2* – codename *Stinger* – means for Linux...

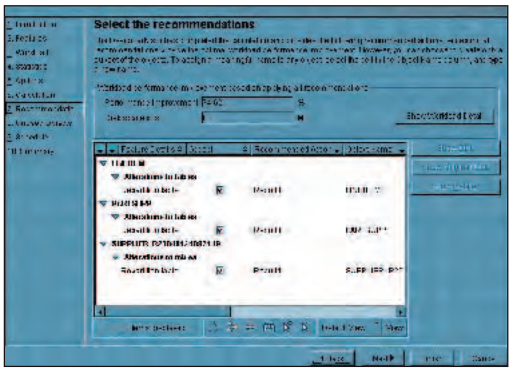
In the enterprise database market, competing on the basis of Open Source code alone is rarely good enough. Reliability is king, followed by scalability and manageability; and failing in any one of these categories is enough to exclude a product from use. Though Free software products like *MySQL* are improving by leaps and bounds, they either don't fully support the database needs of enterprise customers (*MySQL*) or don't come with the kind of bullet-proof support that blue chip companies need (*PostgreSQL*).

The manipulation language behind these database engines (*SQL*) is standardised and thus unable to be used as a marketing advantage for either products. As a result, Oracle and IBM vie for the position of top-dog by making their systems more easily managed, more easily accessible for developers, and, of course, faster than the competition.

Oracle Database 10g was recently released, offering comprehensive support for grid computing. Firing back, IBM has produced *Stinger* – the codename for the next release of its *DB2* database. We spoke to Gary Schneider, the Director of Linux Business Development at IBM's Information Management group, about the motivations behind *Stinger*, how it competes against *Oracle 10g* and Free software databases, and also how it makes the most of the new Linux 2.6 kernel...

LINUX PRO: Will *Stinger* support any new platforms?

GARY SCHNEIDER: *DB2* provides the broadest platform support in the industry, including 22 IBM and non-IBM platforms such as Linux, Windows and Unix. With *Stinger*, *DB2* takes advantage of 64-bit performance in IBM's POWER processors, which lie at the heart of IBM Power Blades (JS20 BladeCenter), and are also offered with pSeries and iSeries servers. As a result, customers can now deploy *DB2* for Linux across IBM's entire eServer family and exploit the 64-bit capabilities of Linux.



***Stinger* will automatically provide recommendations for various day-to-day tasks, and users can choose to accept or reject the suggestions using its easy-to-use GUI.**

LXP: Does the new release officially support the advantages of kernel 2.6?

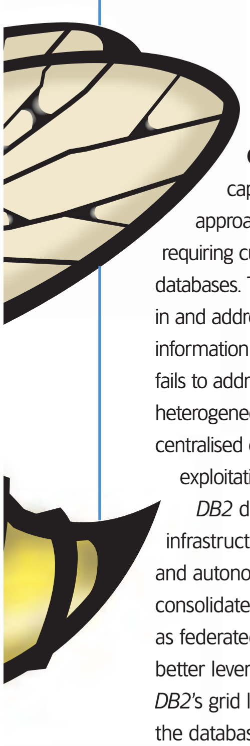
GS: IBM is extending its Linux leadership by supporting the new Version 2.6 of the Linux kernel, which will help IBM's database clusters scale higher and perform faster than before. Support for the new kernel better exploits the speed of 64-bit-ready databases, such as *DB2*, and enables Linux databases to take better advantage of servers that use multiple processors. These multiprocessor servers can be clustered with one another to create powerful Linux clusters.

For example, last year we introduced the *DB2 Integrated Cluster Environment* (*DB2 ICE*) offering for Linux on our xSeries eServer platform. *DB2 ICE* scales much higher and deploys faster than anything else available in the marketplace. *DB2* clusters from two to 1,000 servers – the most in the industry – which can be deployed at a rate of four nodes per hour. Assisting with this rapid deployment is the new *DB2 Design Advisor,* which automatically partitions and optimises large databases on many servers in just a few minutes. Previously, administrators could spend weeks fine-tuning the performance. No other vendor has this capability built into the database.

LXP: Can we expect to see any big performance improvements as a result?

GS: *Stinger* introduces the *DB2 Design Advisor* that automatically maintains, configures, deploys and optimises the database – making it possible to complete jobs up to 6.5 times faster than if done manually. Database administrators can subsequently spend more time with strategic projects, rather than basic database duties. *Design Advisor* also tunes the database on demand as the workload fluctuates, automating any changes to the database structure, as well as backups and restores. The *DB2 Design Advisor* also suggests to DBAs how complex queries can be accelerated, providing the shortest path to the requested information. It does this by learning from the performance of previous information searches, and by collecting, pre-computing and keeping commonly used information at the ready.

LXP: Oracle 10g includes new support for grid computing, giving higher reliability and more



scalability. How does *Stinger* go about tackling the same problems?

GS: *DB2* has delivered true grid computing capabilities long before Oracle and 10g. Oracle's approach to grid computing is database clusters, requiring customers to centralise data into Oracle-only databases. The Oracle-only approach promotes vendor lock-in and addresses only a very narrow spectrum of the information management requirements for grid computing. It fails to address the promise of grid computing to coordinate heterogeneous resources – including those not subject to centralised control – enabling broad-based collaboration and exploitation of existing information assets.

DB2 delivers an information management infrastructure for grid computing that is open, virtualised, and autonomic and meets clients needs both for consolidated, single system scale up and scale out, as well as federated access to diverse and distributed data to better leverage your existing data assets. *Stinger* enhances *DB2*'s grid leadership with autonomic features that enables the database to maintain itself. freeing up a DBAs to focus on more strategic activities.

LXP: What benefits does *Stinger* bring for developers?

GS: With *Stinger*, IBM Business Partners and customers can take advantage of the 64-bit performance of the POWER processors found in our JS20 BladeCenter, pSeries and iSeries eServers. The performance is ideal for demanding workloads such as in Linux clusters.

In general, *DB2* provides the broadest platform support in the industry. By tapping existing in-house skills, developers can lower costs and bring *DB2*-based solutions to market faster. *Stinger* will continue that support by delivering new tools that take advantage of the latest application development features of Java/*Eclipse* and Microsoft .NET (available to *DB2* users even before *Microsoft SQL Server* customers!). *Stinger* also delivers new tools that take advantage of the latest application development features of Java/*Eclipse*, such as *SQL* enhancements with the ability to write stored procedures. This capability enables developers on all platforms to write their applications in the same programming language from start to finish.

Also of interest to developers is support by *Stinger* for larger *SQL* statement sizes, up from 64 kilobytes to 2 megabytes, commonly demanded by third-party applications today. This will make it faster and easier for Partners' complex applications to pose queries to the database.

Finally, in support of creating next-generation geospatial applications on all platforms, *Stinger* customers can take advantage of the new *DB2 Geodetic Extender*. It treats the Earth like a globe, not a flat map, and understands the International Date Line and global time zones. As a result, companies can more easily build more powerful and accurate geospatial applications for land management, asset management, or business development applications that rely on geographical, physical and time-based data requirements.

***LXP:* Will *Stinger* make system administrators' lives easier? Can we expect more automatic configuration and management?**

“*DB2* PROVIDES THE BROADEST PLATFORM SUPPORT IN THE INDUSTRY. *STINGER* WILL CONTINUE THAT SUPPORT BY DELIVERING NEW TOOLS...”

GS: *Stinger* extends IBM's autonomic leadership in the database space, building on our momentum and experience in helping customers reduce the complexity of maintaining database systems. A key competitive differentiator, IBM's approach to database technology is to deliver features designed to simplify, automate or even eliminate many tasks traditionally associated with maintaining enterprise-class database systems. *DB2* lets customers focus on their business, not their technology, and we're freeing up DBAs for more strategic tasks. With *Stinger*, involvement by a database administrator (DBA) is no longer required to periodically refresh *DB2*'s understanding of the data it is managing. *DB2* now learns about changes in the data organisation and adjusts its optimisation strategies accordingly. Competing databases do not offer this level of sophistication and automation – they force DBAs to constantly tell databases how to optimise queries.

Earlier in the discussion, we talked about *DB2 Design Advisor,* which automatically maintains, configures, deploys and optimises the database. *Stinger* offers another series of features to ease the burden on DBAs, and that is new query optimisation technology from IBM's *LEO* (*LEarning Optimiser*) research and development project. *LEO* is the next generation of IBM's query optimiser technology, in which the database automates, simplifies and accelerates queries without human intervention. With *LEO*, *DB2* will now automatically and continually update query statistics about how the database is being used, where it keeps information, and how it is performing. As a result, *DB2* now automatically creates and executes better plans for accessing data.

Additionally, IBM is introducing *Autonomic Object Maintenance*. This new feature automatically performs administration and maintenance functions, such as table adjustments or data back-ups. For example, a DBA specifies what time the database should do its maintenance, the database then considers its workload with the DBA's time suggestion, and automatically performs its maintenance tasks.

***LXP:* How would you say *Stinger* fits into the market place alongside products such as *MySQL* and *Postgres*?**

GS: Businesses that invest in these databases often find that their cost of ownership is not lowered since the technology doesn't have many of the scalability, autonomic and performance features that can lower Total Cost of Ownership. Frankly, we're not coming across these offerings very often – if at all – in competitive engagements, and I don't think any size customer would view them as a viable alternative to databases for running mission-critical applications, like ERP, CRM, Supply Chain, Data Warehousing, Enterprise Content Management. ■■■

KEEP TABS ON DEVELOPMENT

IBM WILL MAKE *STINGER* generally available later this year and offer a simple package for *DB2* version 8.1 customers that want to upgrade. As *Stinger* is still under active development at the time of writing, new features and enhancements are likely to be added before its release. IBM has a section of its *DB2* homepage entirely dedicated to *Stinger* news, and it's worth checking regularly to see what updates have been posted. Check it out at **www-306.ibm.com/software/data/db2/stinger/**

More specifically, IBM has made available a whitepaper discussing the *Stinger* performance enhancements for Linux, available at **ftp://ftp.software.ibm.com/software/data/db2/linux/db2stingerlinux.pdf**

CAN MYSQL RETAIN ITS PERFORMANCE CROWN AS ENTERPRISE FEATURES ARE ADDED?

KEEPING the SPEED

PAUL HUDSON interviews Zack Urlocker, VP of Marketing at MySQL, to find out where the MySQL database system is, and where it's going in 2004...

Though *MySQL* has become the world's most popular database since it was open-sourced a few years ago, it has also benefited greatly from its close ties to the PHP project and also being well-known for high performance. However, the big drive now is to add enterprise-level features such as triggers, stored procedures, and clustering, so we spoke to Zack Urlocker, Vice President of Marketing for MySQL, and asked him to explain just where MySQL is and where it's going...

***LINUX PRO:* How important was the move to *MySQL 4*?**
ZACK URLOCKER: *MySQL 4.x* was important because it brought in many of the core capabilities like transaction support using the *InnoDB* storage engine. We've also continued to boost our performance. I don't have exact numbers, but we generally have 35,000–50,000 downloads a day. So in the last year there have been about ten million downloads. It's a little hard to guess at a total user level, because *MySQL* is bundled in many Linux distros, as well as with Mac OS X, Netware, and other platforms.

There are some organisations still using 3.23 because they don't want to change and don't need the new capabilities, but I think most people have upgraded to *MySQL4.x*. And we expect that trend to continue as we get to *MySQL 5.x*, which adds more enterprise capabilities like stored procedures, views and triggers.

***LXP:* How would you say *MySQL* fits into the database market along with *Oracle* and *DB2*?**
ZU: We serve a different market than Oracle, IBM and Microsoft. Our business is aimed at what we call the 'commodotised' portion of the market. That is, where someone needs a basic relational database, but not all of the high-end features or complexity often associated with products like *Oracle* or *DB2*.

We don't try to have all the features, and that actually makes our product very performant and reliable. And if you don't need those features, why should you pay for them? However, I would add that many of our largest customers often run multiple databases. So, we co-exist with the big three databases. We don't go in and try to replace those products.

Oracle has great grid computing capabilities, and if you need those features, buy them from Oracle: it's a great product. If you want a Ferrari, buy a Ferrari. But if you just need a work-a-day product, we are the Honda of databases. It's fast, reliable and easy-to-use. Many of our customers use *MySQL* for applications where they could not justify the expense of a $50K Oracle license.

***LXP:* You recently switched from the LGPL to the GPL, which caused some problems with other Open Source projects such as PHP. How has this been solved?**
ZU: To be clear, the server was under GPL before, and it continues to be under GPL. The change we made was to make our client libraries under the GPL as opposed to the LGPL, or Lesser GPL. This change was actually made about two years ago in order to correct a situation where some folks were bundling our server with their closed-source application and claiming that because the client license was under LGPL, they could do this. We wanted to make sure that we put an end to this situation by making the client libraries consistent under the GPL.

Our general philosophy is one of *quid pro quo*, or fair exchange. It's fairly simple. If you are building an application under open source, then you can use *MySQL* for free under the GPL. However not everyone wants to release their source code under Open Source; so if you want to be closed-source, then you can buy a commercial license, and we sell those for about an order of magnitude less than closed-source companies. We can do that because of the cost-effective nature of Open Source. We have tremendous savings with Open Source in producing the software, testing it, porting it, distributing our software and so on.

However, although you might think that all Open Source licenses are about the same, there are some incompatibilities with various licenses. For example, some licenses say you cannot have an advertising clause; other licenses say it's OK. So which one takes precedence? In particular, there were incompatibilities with the PHP license, so in December, we issued an exception to the PHP license, making it clear they could use our open source software under GPL with their license. Then we got our lawyers involved, and things slowed down quite a bit, but a month or two later, we issued the general Free Open Source Software (or FOSS) license exception. This makes it clear that *MySQL* can be used in conjunction with any of fifteen or twenty different Open

NEW FEATURES

To what extent are changes in MySQL4 groundwork for future version releases?

ZACK URLOCKER: Sure, 4.1 is groundwork. We are always working on new versions: 4.1 is in Alpha now and it includes some important capabilities including subqueries, prepared statements, unicode support, and some other tuning. We expect this to be beta shortly and will be released for production use in Q3. We very carefully make sure that fast performance is maintained, as that's a key reason why people use *MySQL*.

source licenses like the *Apache* license, Artistic License, MIT license and a bunch of others.

But I admit, in the past we haven't always been clear in communicating our licensing policies, and we're trying do this better. For most people it's not a big issue, but we get taken to task once in a while and I'm glad we cleared it up with the FOSS exception.

LXP: Do many developers in the Open Source community actually submit code to *MySQL*?

ZU: We do get some code contributions, but it is mostly in the form of bug reports or fixes, or features outside the 'core' of the database, like JDBC or .NET support, tools *etc*. After all, the internal database code is pretty sophisticated. But maybe we'll get some good suggestions from the guys who download our database in Redmond or Redwood City. But most of the code is written by our employees and we own the copyright. When we do accept contributions, we ask people to assign over the copyright so there's no confusion about who owns the code or where it comes from.

LXP: Of course, the question is: does it scale?

ZU: We've tested it up to 48 nodes and it scales great. The nice thing is it can run on commodity hardware. It's a 'shared nothing' architecture so there's no expensive shared disk required or storage area network. You can start with just two CPUs or two boxes and get very fast failover. Typically it will detect a failure, like a software crash, network error, hard disk error etc and recover in a matter of tens of milliseconds. The *MySQL Cluster* software will detect and correct errors to ensure that the system is always up and running.

LXP: How many have attended a *MySQL* training session? What's the general feedback you get?

ZU: We also have training and consulting services, certification and so on. We offer half a dozen different training courses in North America, Europe and Asia. We keep the class sizes pretty small and interactive and our training guys are real experts in the technology. So we generally get very high marks.

If you're getting started with a project, getting a team trained is a good way to kick things off. And for partners, consultants, System Integrators, they can get officially certified so that they can also deliver high quality services.

LXP: *MySQL* has been open-sourced for some years now: how has the business model worked out?

ZU: The business doubled in 2002 and again in 2003. But it is off a fairly modest base. But we generated around $12 million in revenue in 2003 and we ran at a break-even basis at the end of the year. We also doubled in terms of the number of employees from around 60 employees to 120 at the end of the year. That's a pretty significant growth rate, especially considering we now operate in 16 countries.

We're continuing to grow by taking care of our customers and I think we're also helped by the awareness people have about Open Source. In particular, there are hundreds of thousands of companies who got the benefit of saving money with Linux compared to closed-source operating systems and now they're thinking about getting an entire open source stack. Sometimes they call this the LAMP stack for Linux/*Apache*/*MySQL*/(PHP/Perl/Python). And sometimes they call it LAMPJ to include Java or Jboss.

But we see many IT organisations now saying if an open-source database is capable of running Google, Yahoo!, UPS or Sabre, then it's good enough for them. And of course, there's always cost pressure: do more with less. And Open Source can help there tremendously. We've also continued to see many developers get started with *MySQL* because they want to learn new skills and become a database expert, but they don't have the budget to use expensive closed source software. Instead, they download *MySQL*, buy a book and they learn on their own.

That's a great way, because then they start using *MySQL* in their organisation, maybe for web applications or departmental applications, and then those run so well, they start expanding the use of *MySQL*.

SUPPORT

What support do you offer for MySQL 4?

ZACK URLOCKER: Since we're in a commodity market, we think its very important to provide high quality service and support. We have many different options available, depending on whether customers need 24x7 support, 30-minute response times or just basic support, say, over email.

> "WANT A FERRARI? BUY A FERRARI! IF YOU JUST NEED A WORK-A-DAY PRODUCT, WE ARE THE HONDA OF DATABASES – IT'S FAST, RELIABLE AND EASY-TO-USE."
>
> ZACK URLOCKER, MYSQL

MYSQL CLUSTER

What problems does it solve?

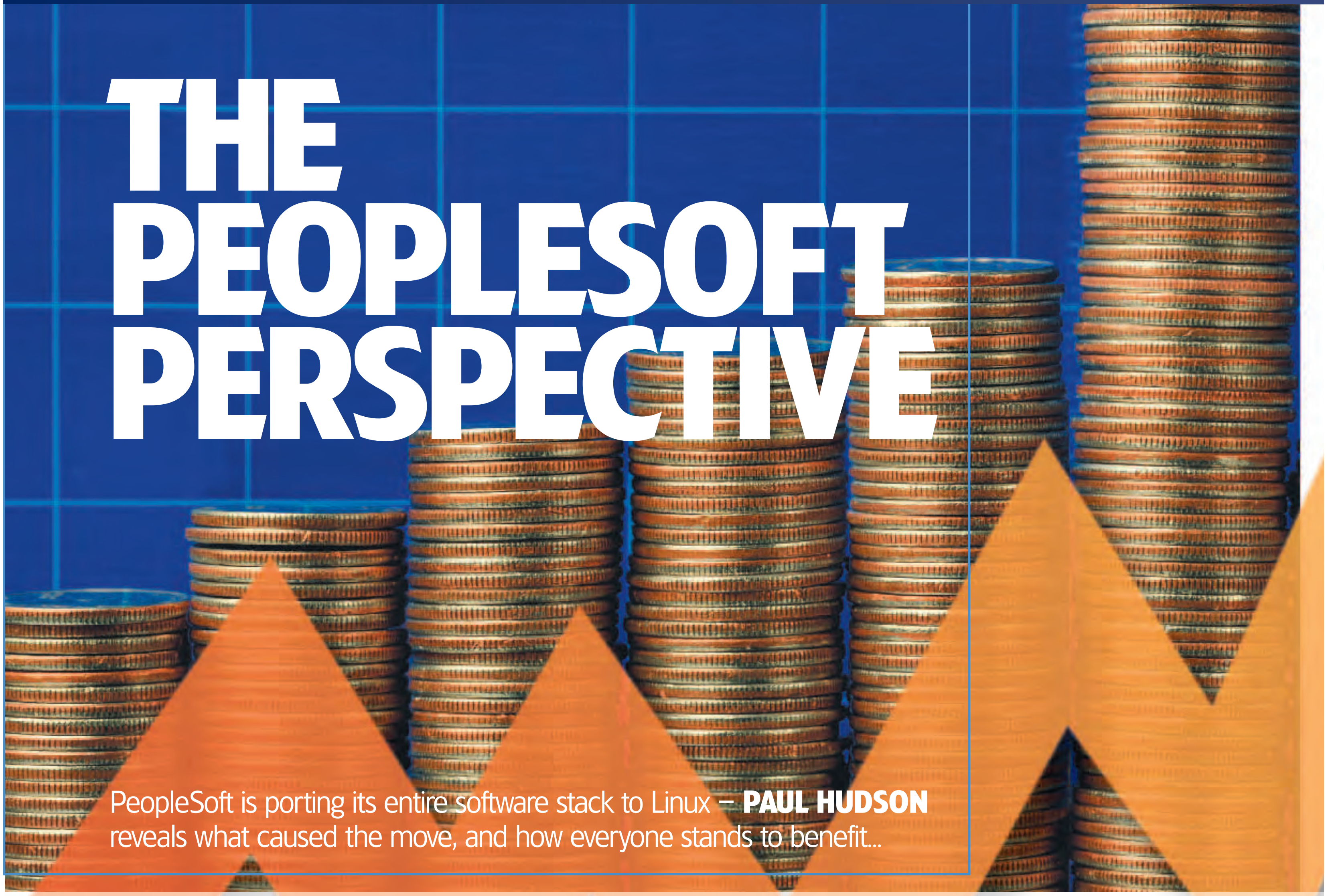
ZACK URLOCKER: *MySQL Cluster* provides five 9's availability. It was technology we acquired from Ericsson, who had been developing a high availability database for the telecom industry. Ericsson spent 75 man-years building this as a closed-source database and we have decided to open-source it under our dual license. So now anyone can have high availability with any *MySQL* application, instead of it just being the domain of the most advanced users. We have some customers using this now in trial and in production applications.

LXP: Do many companies license *MySQL* under the commercial licence?

ZU: We've got around 5 million active installations and 5,000 paying customers at this stage, which is pretty good. We're a small company compared to the big three, but we are growing quickly by serving customers who probably wouldn't be able to afford the big guys.

But we're learning as we go along. There have been cases where we haven't been as good at communicating our policies to our partners or to our customers. And of course, some of the licensing issues around the GPL can be complicated. We didn't write the GPL but it's a bit like democracy. It's not a perfect system, but it's the best one going right now.

One of the things that I think is important is that we build an entire ecosystem around *MySQL*. We're starting to see the emergence of many third-party products and services around *MySQL*, but it took us some time to get there. In the past, the company was too small to help our partners and now we're getting better. Recently, we've seen companies like QuestSoftware announce *Toad* for *MySQL*, which is a great thing. We've got companies creating hot backup tools, application clustering, hosted CRM offerings that use *MySQL*, certified hosting with Rackspace.



THE PEOPLESOFT PERSPECTIVE

PeopleSoft is porting its entire software stack to Linux – **PAUL HUDSON** reveals what caused the move, and how everyone stands to benefit...

After the years of design, development, and marketing, it's good to have a successful product out on the market. Even then, the work is just beginning - supporting that product on the OS it was developed for is no easy task, and usually involves large teams of technicians answering telephone calls from 8am to 8pm. Some would even say that it's the support aspect that's the hardest of all. Why, then, would anyone choose to migrate their software from one platform to another?

In the twenty or so years since it was founded, PeopleSoft has evolved itself into a software powerhouse, producing the world's most popular Human Resources solutions. Back in 2000, it was one of the few companies that truly understood what the Internet heralded, introducing the Pure Internet Architecture. This meant that all of PeopleSoft's applications ran through a web browser on the client-side. As a result, cross-platform client support was working from day one, there were fewer support issues due to the lack of installation, and clients were always working with the most up to date information.

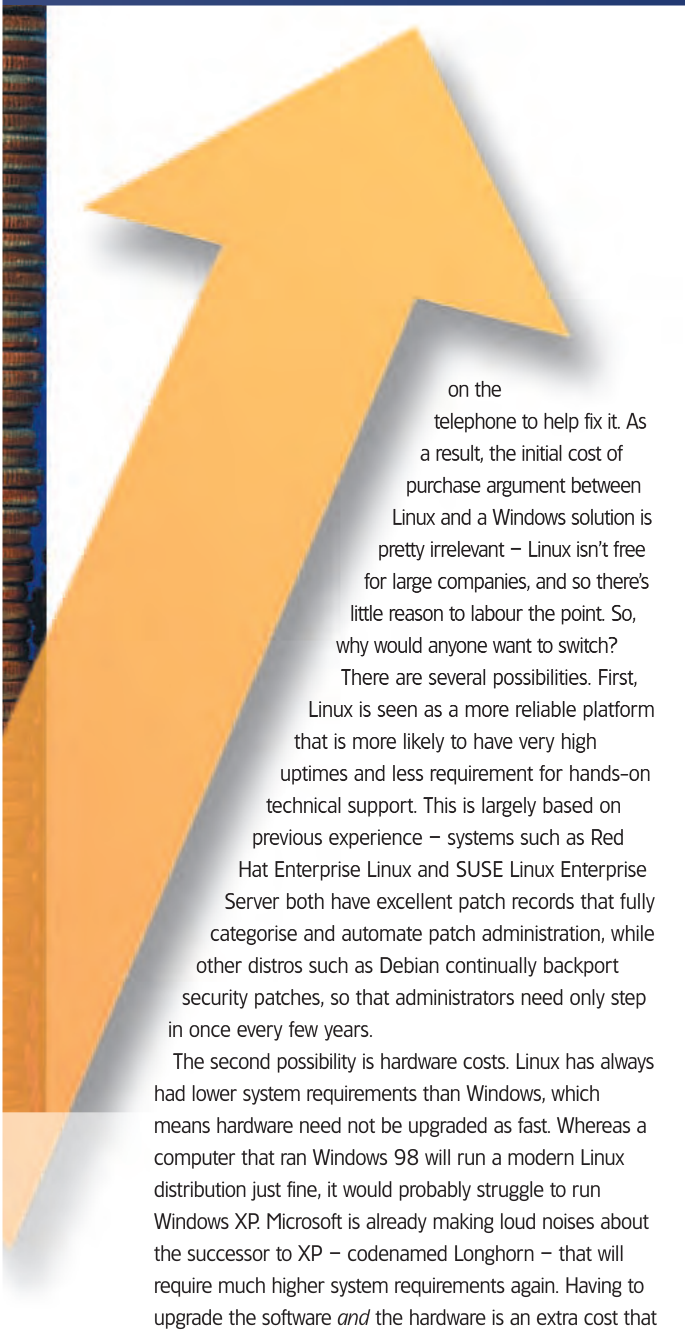
As PeopleSoft software is so ubiquitous in the industry, its decision to port its software to Linux is no small task - the company currently has over 170 business applications, and *every one of them* has been ported to Linux. Not only was this a huge coding effort, but it also required very thorough testing, as well as retraining for technical support staff in order to understand Linux-specific issues.

In this situation, it was no surprise that PeopleSoft chose to partner with IBM in order to help conduct the port smoothly. This was a smart move on PeopleSoft's behalf – other companies have struggled to gain 'mindshare' in the Linux community, with one or two even resorting to purchasing Linux companies in order to buy their way in. IBM, on the other hand, is recognised world-wide as having the industry's biggest commitment to the Linux platform, and also has substantial experience in developing and working with Linux software. Without IBM, PeopleSoft would have had to spend more money on research, testing, and marketing its products, and would likely have not achieved the same level of quality.

WHY PORT?

Although Linux can be obtained and used at no cost, this is rarely a route chose in the enterprise sphere. Companies larger than 50 people would usually rather pay for a Red Hat support contract or equivalent and be safe in the knowledge that if anything were to go wrong, they could have someone

> "WE'RE NOT ANTI-MICROSOFT, WE'RE PRO-CHOICE..."
>
> **CRAIG CONWAY, CEO PEOPLESOFT**



on the telephone to help fix it. As a result, the initial cost of purchase argument between Linux and a Windows solution is pretty irrelevant – Linux isn't free for large companies, and so there's little reason to labour the point. So, why would anyone want to switch?

There are several possibilities. First, Linux is seen as a more reliable platform that is more likely to have very high uptimes and less requirement for hands-on technical support. This is largely based on previous experience – systems such as Red Hat Enterprise Linux and SUSE Linux Enterprise Server both have excellent patch records that fully categorise and automate patch administration, while other distros such as Debian continually backport security patches, so that administrators need only step in once every few years.

The second possibility is hardware costs. Linux has always had lower system requirements than Windows, which means hardware need not be upgraded as fast. Whereas a computer that ran Windows 98 will run a modern Linux distribution just fine, it would probably struggle to run Windows XP. Microsoft is already making loud noises about the successor to XP – codenamed Longhorn – that will require much higher system requirements again. Having to upgrade the software *and* the hardware is an extra cost that fewer and fewer companies want to bear, which provides another compelling reason to switch to Linux.

The third possibility – and certainly one that should not be underestimated – is the desire to avoid lock-in. One of the memos that came to light in the EU antitrust suit against Microsoft was from Aaron Contorer, one of the managers at Microsoft at the time who was writing to Bill Gates. Three statements in that email stand out particularly:

- *"There is a huge switching cost to using a different OS"*
- *"It is this switching cost that has given customers the patience to stick with Windows through all our mistakes, our buggy drivers, our high TCO, our lack of a sexy version at times"*
- *"It would be so much work to move over that they hope we just improve Windows rather than force them to move".*

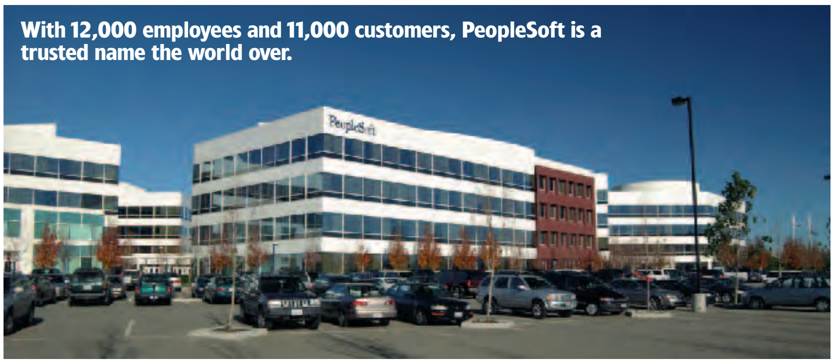
Although what he says may be true, it's not the kind of thing customers want to hear. The fear of lock-in – that once you buy Microsoft you're pretty much stuck with them – shows the advantage of there being several Linux distros. Even if one vendor tried to get lock-in for some reason, you could migrate to any of the other distros with little bother.

PeopleSoft.

IT'S ALL ABOUT MIDDLEWARE

The three porting benefits combined mean that middleware companies such as IBM, Oracle, and PeopleSoft stand to a gain a great deal from porting to Linux. Middleware is so-named because it forms a middle ground between the user and their applications and the OS and its applications, which means that good middleware effectively abstracts the OS functionality. This is important because it means the OS itself is less of a sticking point – to web browsers, a middleware application could be running on Linux, on Windows, on BSD, or all three at once and it wouldn't know the difference.

This means that the software need no longer dictate the OS it's running on, effectively removing the last vestiges of OS lock-in. With PeopleSoft porting to Linux, customers can take their current solution and move to wholesale to Linux with no loss of functionality and no noticeable downtime. It would, however, save money on licensing for proprietary Unixes and Windows. The former of the two has been most popular to date – Linux has made big headway against closed-source Unixes, partly because they are largely interoperable, but also because skills learnt in one Unix can easily be transferred to another.



With 12,000 employees and 11,000 customers, PeopleSoft is a trusted name the world over.

"PEOPLESOFT HAS REVENUES TOTALLING JUST SHY OF $3 BILLION A YEAR, WHICH IS ABOUT AS FIRM AN ENDORSEMENT OF LINUX AS YOU'RE LIKELY TO FIND ANYWHERE!"

By supporting Linux, PeopleSoft offers its customers several immediate benefits:

i Save money on licensing costs
ii Re-use existing training
iii Re-use existing hardware
iv Increase system reliability
v Remove lock-in potential

Clearly these are more than paper benefits; PeopleSoft joins HP, IBM, Sun, Novell, and Oracle in rallying around the Linux platform, offering comprehensive customer support. Perhaps more important, all of these companies have helped the major Linux distros add support for enterprise-level functionality, meaning that PeopleSoft is standing on the shoulders of giants.

POWER TO THE PEOPLE

Strategic deployment

LINUX PRO TALKED TO DAVID SAYED, A marketing manager at PeopleSoft, about why the move to Linux was so important…

***LINUX PRO:* What made PeopleSoft consider moving to the Linux platform?**

DAVID SAYED: In a word, customers. We've definitely seen the interest in Linux amongst our customers shift over time – from the early investigation phase, to the current situation where customers have an official Linux strategy and are deploying our enterprise applications on Linux.

You can really see four distinct phases:

1 Research stage In many companies, this started several years ago with a few people tinkering with Linux on a spare machine. Often these people would become the Linux evangelists both inside and outside their organisations. While they were laying the foundation for Linux in the enterprise, it wasn't until much later that it actually took off.

2 File and print services This is the so-called 'edge deployment', where companies deploy Linux for important but not necessarily mission-critical services: people don't notice if you need to reboot a fileserver as much as they would if you had to take down and bring back up an entire enterprise application or database.

3 In-house web applications This stage really sows the seeds for where we are today because companies realised that they could deploy this open source operating system on low-cost commodity hardware and derive business value from it namely by hosting their own web-based applications. So here, companies got experience deploying databases, web servers and applications in a Linux environment.

4 Commercial applications This is where we are today – mission-critical enterprise application deployment on Linux. Having proved its stability and performance in an enterprise setting, many companies feel that the benefits of Linux outweigh any perceived risks.

Since we spend considerable time engaging with our customers, we've really seen all of those phases and we've been there to offer the necessary support to meet customer requirements. In fact, when we came out with PeopleSoft 8 in 2000, most of the companies we were dealing with were in Phase 3; so, we supported our products on the infrastructure tiers, namely the database and web application server. Also, since our applications are designed to be completely browser-based, customers have always been able run our applications within a Linux desktop environment.

***LXP:* What are the key advantages to customers using PeopleSoft on Linux?**

DS: Customers cite a number of advantages. In some cases, it is lower cost – this could be the cost avoidance of purchasing new hardware, and/or operating system license cost. In others, it is that they get better performance running a Linux-based solution. And for many, they have staff who have skills that can be reused. Any way you look at it though, it is ultimately a cost argument: either bottom-line cost savings or improved productivity, which can also be measured as a cost saving.

***LXP:* What challenges were there in the migration to Linux?**

DS: PeopleSoft has always developed cross-platform products. We support everything from Windows to the IBM z/OS mainframe operating system. That made us well placed to add another operating system to our set of supported platforms, especially one that is very similar to Unix. Therefore, the biggest challenge wasn't so much in development, as in verifying that the end-to-end system with 170 different applications ran correctly, ensuring that any third-party products were in fact supported on Linux, and ensuring that our support organisation was fully trained and ready to handle Linux questions.

We announced our plan to support Linux on the PeopleSoft business logic server (application server) in May 2003, and that we were partnering with IBM to do so. While we had Linux experience from supporting other tiers on Linux, we felt that IBM had not only a deep understanding of Linux across different platforms, but also the industry's strongest commitment to Linux. THe partnering enabled us to get to a completely Linux-based solution faster, since IBM had a wealth of Linux experience to help us to iron out kinks along the way.

***LXP:* Has customer feedback been positive?**

DS: Customer response has been tremendous. In actual fact, immediately after our May 2003 announcement, we received unsolicited calls from customers who were wanting to become early adopters. We worked with several of them to ensure an even higher quality product. It is interesting to note that many customers are seeing this as a cost-effective way of adding additional PeopleSoft applications to their existing PeopleSoft footprint. So the rest of their infrastructure may not in fact be Linux-based – since it was already there – but the new application they license is being deployed on Linux.

David Sayed, PeopleSoft.

***LXP:* How much of your software range runs on Linux already?**

DS: When we made the announcement, we committed to deliver all of our 170 applications on Linux by the end of 2003. We met that goal, but in the interim also acquired JD Edwards. So, all PeopleSoft Enterprise (formerly PeopleSoft 8) applications run on Linux and just last month, we announced the availability of PeopleSoft EnterpriseOne (formerly JD Edwards) applications on Linux. So the only product-line that doesn't support Linux today is PeopleSoft World (formerly JD Edwards World), which is specifically designed for the IBM i-Series.

***LXP:* Does PeopleSoft plan to contribute much back to the Linux community?**

DS: I mentioned partnering with IBM earlier, and one of the reasons for doing so was to leverage IBM's commitment and stature in the Linux community. As the first enterprise application vendor to make such a deep commitment to Linux, we feel that we can provide useful feedback to the community through IBM.

This is a model that has worked well for us in the past with database vendors. There are sometimes features that we would like to see in a particular product offering that have wider appeal than just for PeopleSoft applications. That feedback is valuable to the database vendor and we find that new features to address our needs are rolled into new releases.

Now, Linux isn't a database, and there is no 'Linux vendor' as such; but we do feel that partnering with IBM enables us to provide that feedback to the Linux community, which will in turn make the platform go from strength to strength.

WIN-WIN SITUATION

We've already seen how customers win by migrating their PeopleSoft deployment to Linux, but PeopleSoft has a lot to gain also. First, PeopleSoft smooths its production process – the company already develops for Unix and Windows, and, because Linux is very similar to Unix from a development and support standpoint, the more customers that can be moved from Windows to Linux, the more PeopleSoft can consolidate on the Unix platform. Though there is a saving to be had from being able to reduce the number of Windows licences used internally, the much bigger saving is in resources because people can be more productive.

Unsurprisingly, the Linux community also wins – such strong support from PeopleSoft yet again debunks the myth that Linux has no place in the enterprise, and eliminates one more excuse from companies who have yet to switch. The question is decreasingly *"why are you planning to switch to Linux?"* and increasingly *"why are you NOT planning to switch to Linux?"* PeopleSoft employees 12,000 people, has over 11,000 customers across the world, and has revenues totalling just shy of $3 billion a year, which is about as firm an endorsement of Linux as you're likely to find anywhere!

One question that still hangs in the air is whether we'll see PeopleSoft giving much back to the community as a result of this move. However, whether or not code is handed back, there are at least two fringe benefits. First, other large vendors have implemented certification programs with the major distros to encourage customers to move, which helps fuel Linux adoption – you might say that some CTOs hadn't heard of Linux until IBM started mentioning it. Second, PeopleSoft has partnered with IBM to produce this port, which should re-enforce to IBM that its Linux strategy is succeeding, thereby encourage them to continue it. Either way, the community comes out a winner, and should welcome PeopleSoft's support with open arms.

STILL SOME WORK TO GO

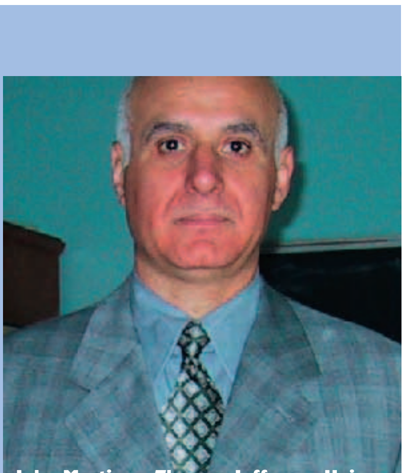
As the PeopleSoft products roll off the press and finish quality assurance, some companies may be less-than eager to make the move, despite knowing it would benefit them. The problem here, of course, is that the first release of a product is usually a little flaky as it hasn't received a thorough testing. This is mitigated to a degree because of Red Hat's continuing five-year minimum support plan for RHEL – Enterprise Linux 3 has been out for six months now, and will continue to be supported in at least four years' time, which means that PeopleSoft can commit safely to the platform and make sure it works solidly there.

Again, the IBM partnership has meant that PeopleSoft built on IBM's experience and avoided the most common errors – poor installation, reliance on old or inconsistent libraries, *etc.* Perhaps the most important aid to early stability, however, has been PeopleSoft's extensive use of alpha and beta testers, as well as early adopters who are eager to make the move to Linux. Not only has this given PeopleSoft valuable insight into what customers need to help them get set up and working, but it will also have helped iron out many remaining bugs in the system leaving it safe for even the most mission-critical of systems.

EARLY ADOPTERS SAY "YES"

Economic migration

LINUX PRO ASKED JOHN MARTINES, Assistant Director of Enterprise Systems at Thomas Jefferson University, why PeopleSoft on Linux was right for academia...



John Martines, Thomas Jefferson Uni.

***LINUX PRO:* What made you want to migrate to Linux?**

JOHN MARTINES: The main reasons for migrating my PeopleSoft environment to Linux are reliability, cost, and team experience. We have found that installing and maintaining Lintel boxes running Linux is more cost-effective – they just run! Our servers were configured back in December of 2003 and we have had no need to reboot them, and they have been tuned specifically for our PeopleSoft needs.

***LXP:* Had you already deployed Linux elsewhere on your network?**

JM: Yes, we run our university mail on Linux. There are over 35 Linux servers in our server farm. We have a relatively small Linux server processing tens of thousands of emails a day.

***LXP:* What problems did you encounter while deploying?**

JM: We have not encountered any major problems. We did use PeopleSoft to setup our PeopleSoft Portal 8.8, running the PeopleTools 8.4 tools, and there was very little assistance that was needed from PeopleSoft to configure our Linux environment. We are pleased to say that our Employee Self-Service Project is scheduled to go live next week, and we do not foresee any issues.

***LXP:* Has the move helped save money, or has it cost more?**

JM: One of our challenges in our University environment and being at the end of our Fiscal Year, we did not a lot of money available to spend towards this project. By rolling out Linux and utilising servers already available to us, we were able to focus our finite dollars on functional and application level consulting versus on hardware or technical assistance. Our cost for the rollout of this project was very manageable – and palatable to our Senior Management team.

> "WE HAVE A RELATIVELY SMALL LINUX SERVER THAT PROCESSES TENS OF THOUSANDS OF EMAILS A DAY."

To give you an idea quite how important this move is for Linux, consider this: as more and more companies go ahead and migrate to PeopleSoft on Linux, what we'll have is pervasive – if behind-the-scenes – deployment of Linux in the workplace. Linux has yet to get much of a foothold in the Human Resources industry, but, equally importantly, neither has Microsoft. By putting its weight behind Linux, PeopleSoft is enabling hundreds-of-thousands of users to rely on Linux in their workplace, which is a real triumph for the OS and is one more barrier broken down between full Linux adoption.

We are now at the point where companies who don't show commitment to Linux are in the minority. Going forward, we fully expect PeopleSoft to be joined by several of the remaining few 'Linux hold-outs' before the year is out, and, in the meantime, PeopleSoft's customers will get some serious benefit from the switch to Linux. ■■■

infosecurity
EUROPE
27–29 April 2004
OLYMPIA, LONDON
www.infosec.co.uk

SHOW REPORT

INFOSEC 2004

NICK VEITCH braved the madding crowds at Olympia's Grand Hall to bring you back the Linux-related news and gossip from Europe's biggest information security event.

Billed as Europe's number one security event, the Infosec 2004 event was certainly the biggest, dedicated security show we've ever seen - even bigger than last years event, and promising to be one of the most important events on the calendar for those involved in IT.

In the same venue as previous years, the Grand Hall at Olympia, the Infosec show has grown rapidly over the last few years, reflecting increased awareness of security problems and the number of companies and organisations concerned with this sector.

Initial (unaudited) figures suggest a 14 per cent increase in visitor numbers from last year, many of these from overseas.

DTI

One of the highlights of the show was the publication of the Department of Trade and Industry's Information Security Breaches Survey 2004, which was delivered on the opening day of the show. This massive survey aims to plot trends in the use of IT throughout industry, as well as identify common vulnerabilities and problems. The results are in some cases surprising.

Firstly, the way business uses technology has changed dramatically since the last survey in 2002. Many of these changes are only to be expected – a growing reliance on email, more and more businesses having some kind of web-based customer interface, and an increase in outsourcing IT operations.

Wireless networks in business have grown from 2 per cent to 34 per cent overall, with the most uptake being with large businesses, a staggering 75 per cent of whom now have a wireless aspect to their network. Remote

access to company IT services has also doubled in the period, with over half of businesses having this capability. It is perhaps unsurprising then that greater connectivity has resulted in a comparable growth in reported security threats, due to the proliferation of targets and the increased potential for abuse.

On the issue of viruses, it is reported that the average UK business receives around 20 viruses per year, with larger businesses averaging one per week. On the plus side, this is an area where business is aware of the risks, and practically every company surveyed was able to affirm that they had an up-to-date anti-virus implementation.

Unfortunately, business doesn't seem to be so prepared in other areas. The number of malicious security breaches is up by 50 per cent, and 25 per cent of businesses had an incident that resulted in systems failure or corruption of data.

The costs of such breaches have also risen. The average cost of a companies most serious security breach is no £10,000, rising to £120,000 in larger businesses. One of the most significant contributors to the costs is the impact on systems availability and disruption to the business.

The lessons to be learnt are quite clear in the rest of the report. Although the complexity of the IT infrastructure has increased, with new functionality and extra productivity bonuses, the ability of IT departments to secure and control this environment has not kept pace. Stunningly, the survey revealed that half of wireless network implementations have no security controls, for example. In spite of virtually every IT magazine exposing the perils of unsecured wireless networks, and a proliferation of websites identifying where such access can be found, only 50 per cent of companies using the technology have bothered to secure it.

Many companies now seem to have some sort of disaster recovery plan, but a surprising number of them have never tested it! Obviously, the time to discover flaws and hiccups in the system is before your business starts relying on it...

The survey goes on to suggest that training and knowledge of IT staff in the vulnerabilities of the new technologies there are deploying may be responsible. Of the businesses surveyed, 80 per cent of IT staff had no formal qualifications in security. This perhaps isn't too surprising, since security training hasn't been a 'big thing' until the last couple of years.

SEMINARS

INFOSEC HAS ALWAYS RUN a thread of seminars and workshops, but this year there were several threads, and a lot of timely topics too. You can often be forgiven for thinking that most of these are largely self-promotional exercises, but there was a lot of common sense being talked in the small seminar rooms at the back of the hall, and for many of the visitors I spoke to, it was worth the trip to infosec just to attend them. Some of the more adventurous demos involved live-hacking attempts, but in the main they were just good, solid old-fashioned presentations on the perils that await the unwary in today's over-networked IT landscape.

To summarise, the report really outlines what we should already know – the IT world is changing as always, but perhaps at a more fundamental and rapid rate than before in terms of the needs and methods of accessing information. Change brings risk, and change is responsible for most IT failures. Be sure that whatever technologies you plan to implement, they are also incorporated into your security and disaster recovery plans.

ASTARO

Astaro is probably one of the better-known flavours of security-hardened Linux available. From early 2000 when the first version was released, it has continued to build on the idea of an all-in-one security device distribution.

With built-in firewall, intrusion prevention, anti-virus and anti-spam capabilities, it also now includes web filtering and VPN functionality. It's not surprising that Astaro is very popular, and there were plenty of appliances at the show based on the software. We'll be taking a longer look at Astaro in *Linux Format* issue 56 next month.

SOURCEFIRE

Avid readers of *Linux Pro* may well remember our interview with Marty Roesche almost a year ago. As well as being the pretty famous author of the excellent *Snort* utility, he is also the founder of Sourcefire, a company set up to make security appliances.

In the interview, Marty mentions that he is working on a new technology for passive network analysis that will identify patterns and behaviours to make more sense of the raw packet information. That technology is here! *RNA*, or *Real-time Network Analysis*, aims to provide up-to-the-second, contextual data on a network, including assets (such as connected devices), behavioural data

»

(traffic types), vulnerabilities evident on the system and assets which have changed (*eg* servers opening new ports, *etc*).

While previously the problem with packet monitoring was the sheer bandwidth of information, the idea behind *RNA* is to intelligently decide which information is important, reducing the management overhead. *RNA* is certainly an exciting technology and we were delighted to find that Sourcefire now has offices in Europe. You can check out the sourcefire website (**www.sourcefire.com)** for more information on *RNA*, but rest assured we will be covering it in more detail in a future issue.

SMOOTHWALL

There can't be many readers of *Linux Format* and *Linux Pro* that don't know of SmoothWall. Originally started as an Open Source security project, the GPL version of the software is still going strong, mostly thanks to the development effort now funded by sales of the corporate version of the software.

This is the first time SmoothWall has attended the Infosec event, but they generated plenty of interest, particularly in some of the recently announced additions to the Smoothwall Corporate Server solution and enhancements to other products.

One of the more interesting recent announcements is the latest release of *SmoothGuardian*. This is a web content-filtering module for the corporate product, and offers a few genuinely new features as well as plenty of speed improvements over previous versions. One of the more forward-thinking options is to have time-based policies, so for example, users are permitted to visit entertainment sites out of office hours. This would certainly help enforce the less draconian AUPs that are becoming more popular throughout business.

Another recent release is *SmoothTunnel*. The major inclusion here is the support for Layer 2 Tunnelling Protocol (L2TP). The benefits of using L2TP are that it doesn't choke on time-critical streams such as VoIP and streaming media as *SSL*-based solutions can. It's also surprisingly cheap and easy to implement, as an L2TP client already ships with the Microsoft Windows XP operating system.

Nick Veitch chats with SmoothWall MD George Lungley about the future of Linux information security.

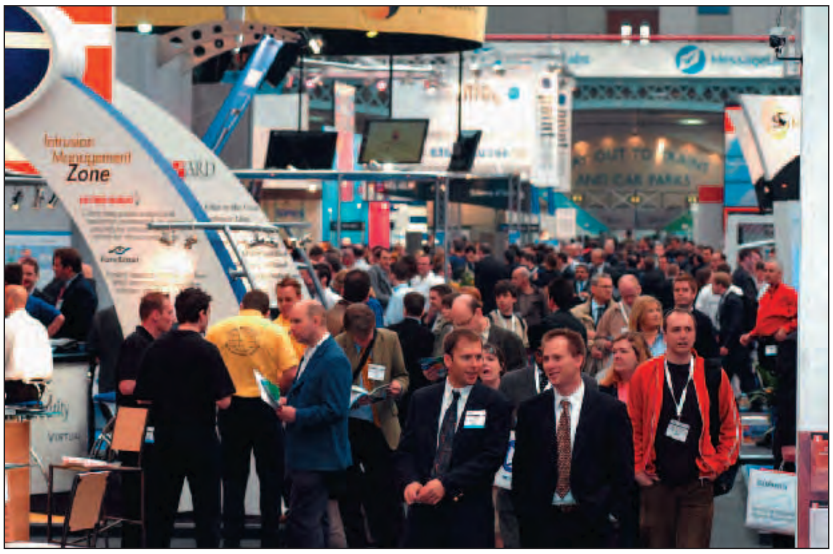
MESSAGELABS

As one of the biggest mail-services companies, it was no surprise to see MessageLabs at the show. As the DTI survey clearly indicates, many companies seek to reduce costs and worry by outsourcing various parts of their IT framework, and this is perhaps one of the more seamless parts of network technology that is easily incorporated into this strategy.

MessageLabs featured in *LXF*'s cover-feature last issue as an option for virus and spam prevention; it also provides services for web content filtering too. With spam still undergoing exponential growth, it could be more economical to outsource this rather than take the server hit that goes with processing 50 per sent more mail.

BIOMETRICS

The latest hot thing in the world of security, judging by the companies exhibiting at Infosec, is biometrics. Thanks to the



Plenty of big names, and encouragingly, many many new exhibitors this year.

US's new policy on passports and visitors' visas, it's a term that more people are aware of at least, and the range of products and support for them is growing rapidly. Solutions on display at the show ranged from a number of companies with fingerprint readers in various forms (in USB keys, built into devices such as keyboards and so on) to more complicated affairs including a 3D facial-recognition system, which could reputedly tell identical twins apart .

ActivCard exhibited at the show, with various fingerprint-reading devices. ActivCard supports Linux clients for its biometric and card/token based security solutions – find out more at **www.activcorp.com/**.

IN SPITE OF PROPHECIES ABOUT LINUX VIRUSES, MOST VENDORS PROMOTED THEIR WARES TO THE MAJORITY SUFFERERS – WINDOWS USERS

Perhaps a more logical place for fingerprint testing is combined in the traditional keyboard. That was the system that well-known HID manufacturer Cherry was demonstrating at the show, though nobody at its stand could tell us whether there was any software support for the device on Linux at the moment.

ANTI-VIRUS

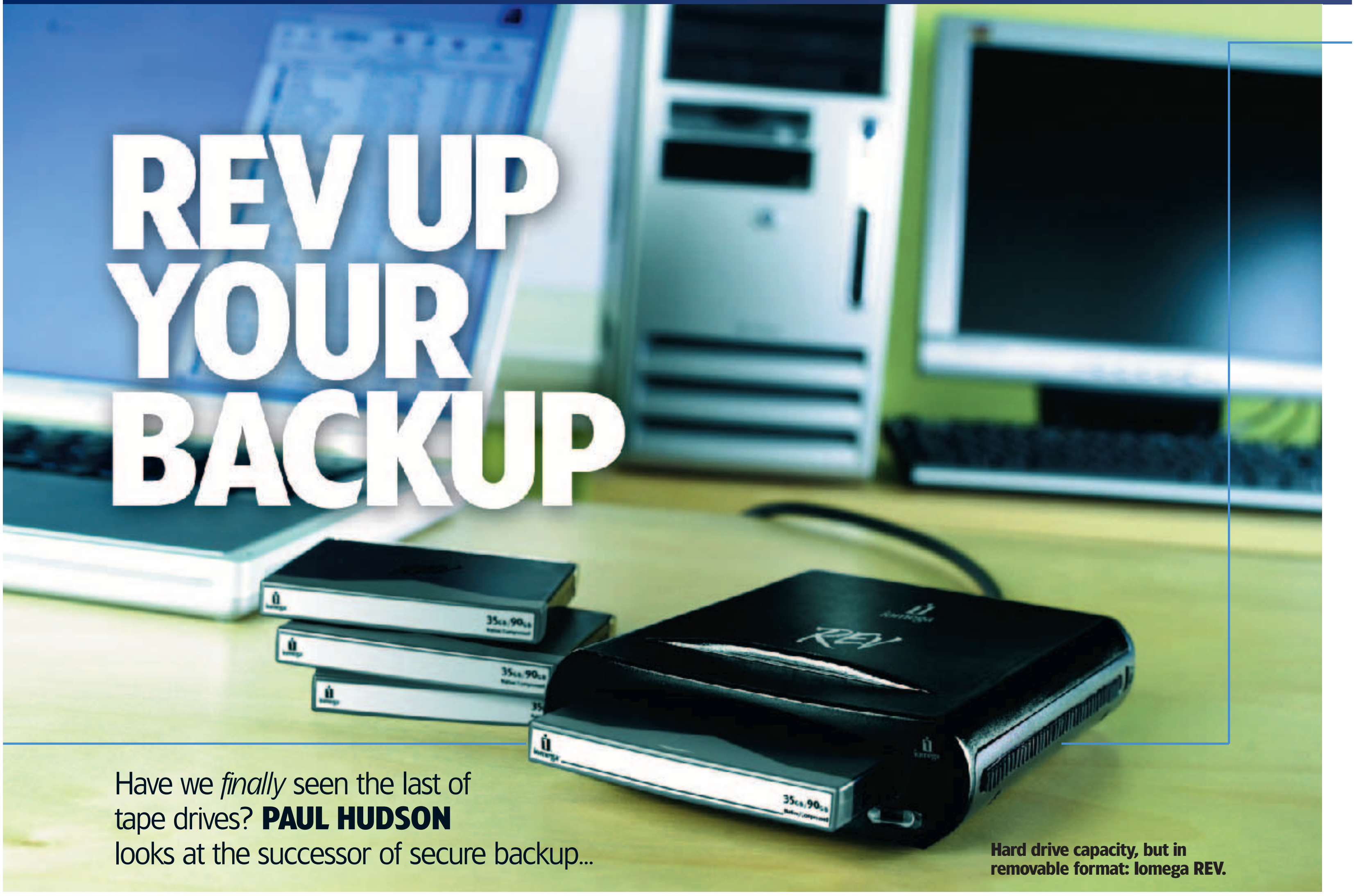
With the DTI report confirming that 70 per cent of businesses have suffered some sort of disruption due to virus attacks, it wasn't surprising that a large number of anti-virus software vendors were in evidence. Although a smattering of these offer Linux clients, and in spite of numerous prophecies about a sudden plague of viruses being unleashed on the Linux platform, they were largely content to promote their wares to the majority sufferers – Windows users. That's not to say that many of them haven't become smart to the use of Linux in heterogenous networking environments. As Linux is often used on fileservers, Linux virus-checkers can filter out infected files before the Windows clients get to them. Companies like Sophos and Kaspersky have Linux client versions that are suitable for this, and mail-scanning purposes.

One of the most popular anti-virus tools on Linux is *ClamAV*. By nature of being an Open Source initiative, the developers presumably couldn't afford the time and expense of attending the show. Perhaps this is one area the organisers can look into for next time – although the show was pretty packed, Open Source initiatives form a valid and essential part of the security community. ■■■

DATE FOR YOUR DIARY

INFOSEC EUROPE 2005
Dates: 26–28 April 2005
Venue:
Grand Hall, Olympia, London
Time:
Tuesday 26 April 10.00-17.00
Wednesday 27 April 10.00-17.00
Thursday 28 April 10.00-16.00

REV UP YOUR BACKUP



Have we *finally* seen the last of tape drives? **PAUL HUDSON** looks at the successor of secure backup...

Hard drive capacity, but in removable format: Iomega REV.

Not so many years ago, Iomega revolutionised the storage market by releasing its Zip drive – portable disk drives that everyone could use and had high capacity for their time. Although the latest Zip drives hold 750MB of uncompressed data, that's still not really enough for demanding users. Even CD writing is beginning to wane as DVD writing waxes – because multimedia is increasingly about video editing, 8-megapixel digital cameras, and movies, movies, movies.

Of course, there is another market out there that is constantly screaming for more space: tape backup technology has not advanced as fast as the industries using it, whereas hard drives have. As a result, even an eight-tape caddy holding a full array of 40GB tapes only holds 320GB, which isn't that much, considering computers ship with 100-160GB drives as standard. In the arena where data is critical – which is pretty much every company worth its salt – backups are usually done every night so that the worst that can happen is a 24-hour setback. In this situation, the procedure is usually performed overnight, starting usually when everyone has knocked off for the night, and hopefully finishing before 9am. However, this highlights the second limitation on backups today: they are just taking too long.

So, what was needed was a way to backup more and do so faster, and Iomega has once again come to the rescue with a product designed to dazzle and delight: *Iomega REV*.

SPACIOUS AND SPEEDY

As with its predecessors, Rev is disk-based and so benefits from all the standard disk features – high performance, random access capability, and long life. Each disk holds

IS IT 35GB OR 90GB?

EACH DISK HOLDS 35GB OF UNCOMPRESSED data, but Iomega usually refers that figure as well as the estimated amount of compressed data it can hold – 90GB. That figure gives an approximate compression ratio of 2.6:1 which is higher than the 2:1 that most tape drive vendors quote, and substantially higher than the 1.5:1 most of us actually achieve. Is this because of some new technological advance Iomega has made in Rev?

Well, no – it's either just wishful thinking, or Iomega has chosen very specific usage scenarios. For example, plain text usually compresses at 2:1, but *OpenOffice.org* docs don't compress at all as they are already compressed. Similarly, bitmap files might compress at 9:1, whereas JPEG picture files will barely compress at all as they are also already compressed. It's this range of compression possibilities that leads most companies to average it out at 2:1, but your mileage may vary. Another factor is that Iomega uses the System International (SI) scheme of measuring capacity that puts a Gigabyte at 1,000,000,000 bytes. While technically correct, it does mean that Iomega's meaning of 35GB is, to everyone else, just 32.5GB. This remains standard in the storage industry, which means a 35GB Rev drive would backup a 35GB hard disk if such a beast existed.

35GB as standard, although compression will likely take this to between 70-90GB depending on your content. The drive itself works similarly to its predecessors also, in that you plug it in to your computer and it looks and works like a standard disk – you can drag-and-drop files in there, format it, *etc*, as if it were a hard drive.

What stands out most, however, is the performance: you can expect up to 25MB/s from Rev in sequential data transfers, which is blindingly fast. At initial launch the drives will be limited to USB 2.0 support, which may leave its speed at a relatively crippled 15MB/s, depending on the quality of your USB 2.0 controller. This will change, however, as more support is added for other access methods – SATA access is expected later this year, and should support Rev at full speed.

Given that Gigabit networking is pretty much the norm now, and that a good network is able to transfer in excess of 100MB/s over the wire, there's quite a bit of slack left over. In comparison, current tape drives tend to crawl along at under 10MB/s, which puts increasing pressure on the network administrators to get the nightly backup done before work starts the next day.

Although 25MB/s may not sound all that much, you should also know that Rev allows on-the-fly encryption using the new AES standard, even while compressing at up to 2.6:1. What is particularly interesting is the possibility of Rev autoloaders, which is something that German manufacturer BDT is looking into. The idea here would be to bundle several Rev disks together into one removable caddy so that backups of 300GB upwards can be done in one shot and at lightning speeds.

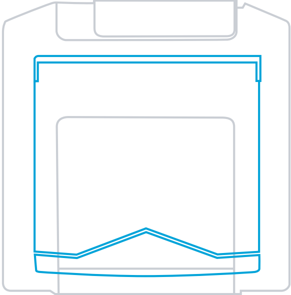
For now, Rev only supports 90GB a time, which means that large backups will need to be done across discs by hand. This isn't quite so bad as it seems, as the exceptional performance means an entire backup can be done in about an hour. More importantly, an entire *restore* can be done in the same time, which means that downtime as a result of disaster is slashed – something that's bound to greatly please administrators.

MAXIMUM RELIABILITY

Although the infamous 'click of death' problem plagued Zip drives for a short while, that's long gone and hopefully never likely to return. Newer Zip models were free from this problem, but that hasn't stopped Iomega raising the bar even further this time – UDF, the data format used on DVDs and other optical media, is used for error correction, and there are even air filters fitted inside each disk to make sure that it remains dust and dirt-free even after many years. Iomega's goal was to get as close as it can to the clean-room-like conditions in which hard drives are manufactured, so that each disk can be used millions of times before they start to degrade.

Once you add that to the fact that Rev already does automatic drive head cleaning, and also that all the sensitive components are kept in the drive unit as opposed to on individual disks, and it's no surprise that Iomega are quoting a 30-year shelf life for Rev disks. Having said that, only a thin metal strip separates the hard drive surface from the outside world, which does seem a little flimsy.

To make it quite clear that it's serious about enterprise backup, Iomega has also stated that Rev drives will be shipping with full Linux support out of the box by the end of the year. As they work with UDF and are essentially external disk drives, there's already unofficial support being produced by the community – SourceForge hosts the most popular project at **http://sourceforge.net/projects/iomrrdtools/.** Even though a community driver would allow support for Rev to be bundled with every distro as standard, it's unlikely Iomega would support companies using anything other than its official driver set.

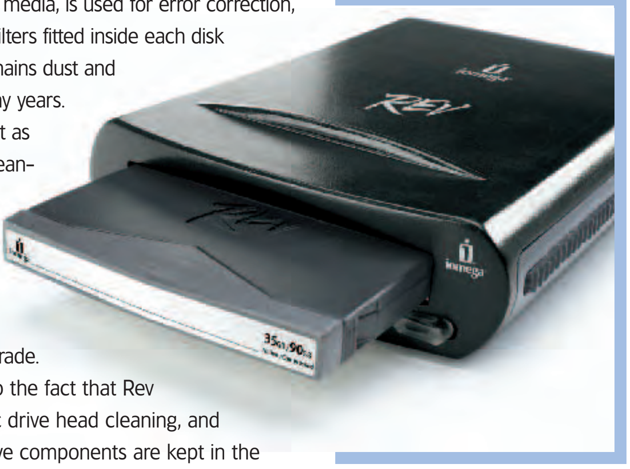


Old Zip 100 (grey) and new Rev drive cartridge (blue): improved performance and capacity, but a smaller form-factor – 10 x 92 x 74mm.

LITTLE WONDER

How much space does a Rev drive take up?

FOLLOWING ON FROM THE Zip disks, the new Rev media remains fairly small and lightweight. It's about 25% smaller in width and length than the Zip disks, but, at 1cm, is almost twice the thickness. That's still pretty tiny, though, and shows quite how far Iomega has come since the original Zip 100 disks – the capacity has gone up by 350 times, but still fits snugly in your shirt pocket. Equally importantly, the Rev media weighs just 73 grams, which, despite being heavier than the original Zip disks, is still negligible.



SUITABLE FOR THE ENTERPRISE?

Though there's no doubt this is a huge leap forward for data storage, some question whether it's enough of a revolution to capture the hearts and minds of end-users. Speed isn't a problem, as the Rev drive out-performs everything else on the market, and is likely to continue doing so for some time. Capacity, however, is where it falls down: Iomega has relied heavily on the home-user 'just change the disks' thinking that led to success with Zip. At the same time, tape autoloaders the world over come with ten times Rev's capacity – during an overnight backup, no one wants to hang around and change the disc every half-hour.

Even without autoloaders, there's another problem – 35GB is not a 'standard size.' In the same way that 515MB of RAM isn't 'standard' (you either have 256, 512, 1024, *etc*), you won't find any 35GB hard drives out there. Annoyingly, the standard in low-end SCSI drives is 36.5GB, which means that a Rev disc falls *just* short of backing up a full drive. Clearly Iomega has a cut-off point to the capacity of its discs, but if the company could have squeezed just that little bit more out of it, everyone would have been better off.

On the flip-side, the biggest advantage to Rev is the extra reliability it brings – tape technology has been around for more than 30 years now, and although clearly things are more advanced now than your early cassette copies of Abba's *Greatest Hits*, there's only so far the technology can go before a dead-end is reached. From this perspective, Rev's long life is highly desirable and probably even eclipses its performance in terms of importance.

For home users and enterprise workstation use, Rev drives are definitely the way forward, and should ease sharing of data in the corporate environment. Although it would be nice to use the network for everything, until we get Terabit networking, it's not really feasible. For centralised systems, Rev is ideal for incremental backups that fit into one disk, and given that many companies practise the 'incremental backup every night, full backup once a week' philosophy, this means that Rev and tape could live together to give the best of both worlds.

If it's true that a full autoloader system is already on Iomega's drawing board, then Rev is set for an even wider adoption and, at last, would mean we're finally going to see the last of tape drives... ■■■