

Bigger, better, more...

Well, it seemed you wanted more tutorials, and you've certainly got them. You've got more pages too and an extra CD for those not getting the DVD version. And we've increased the range and scope of the tutorials – from short informative ones like the *SySRq* and *ext3* tutorials to the likes of *Kylix* and *partimage* that you can really get stuck in to.

We realise that whoever you are, and however long you've been using Linux, there are always new things to learn, and we try and reflect that in our tutorials. If you've got a subject you'd like to see us cover, just drop us a line at linuxformat@futurenet.co.uk.

Our two features this issue have a highly practical bent too. VOIP is a fast-growing industry, and while there is some doubt over standards, Linux has both of the major contenders covered. Find out how you personally, or you as a business, could save money with this enabling technology.

And if, like many, you wake up sweating in the

middle of the night, after a terrible nightmare where your server goes down in the middle of the night and you're bankrupt by daybreak, perhaps you'd better take a look at our guide to High Availability and get a good night's sleep for a change.

You'll probably also have noticed that there are now two CDs with your favourite mag (unless you've already got the supersized DVD version). So we can indulge those of you hankering after the latest distros, we've chosen a great, brand new version of Linux. They may be changing their name (see the news pages), but this is still a great distro for new users or those migrating from Windows – check out the review on page 28 or play with it yourself – the install guide starts on page 102.

Of course, there's much more than that in this huge issue, so be sure to check out your regular favourites and our reviews of everything from library hardware to the latest distros.



Nick Veitch EDITOR

LINUXformat Aims of the magazine

Linux Format is a magazine dedicated to Linux and the Open Source community. The aims of this magazine are quite simple:

- » To promote the use of Linux by providing friendly, easy to follow guides to installing and using this operating system.
- » To help our readers get more out of their Linux experience, through our tutorials, features and advice pages.
- » To provide Linux Users with accurate and unbiased information.

In the future you will have to put on a tie to answer the phone.
VOIP for Linux explored **p48**



Don't have nightmares, have a high availability Linux server **p54**



Our expanded tutorials pages put a world of Linux in your hands **p67**

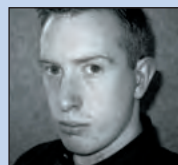


Meet Linux Format's team of writers...



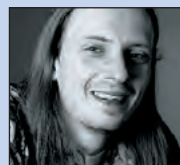
Richard Smedley
Discovered under a bush on the LXF allotment, only Rich2's Linux experience saved him from composting.

David Cartwright
Veteran journalist and Linux consultant, he knows his stuff when it comes to real-world Linux usage.



David Coulson
An LXF regular, David is a networking and security guru who has no problems with his laptop, thank you.

Hoyt Duff
Fishing pier proprietor Hoyt spends his spare time installing Linux on anything that stays still long enough.



Richard Drummond
As well as writing our Java series, Rich co-ordinates most of the reviews in the mag.

Biagio Lucini
He knows his devices, and his filesystems, so don't miss devfs exposé or his ext3 tutorial this ish.



Jono Bacon
Founder of Linux UK and big KDE exponent, Jono is studying Multimedia at Wolverhampton Uni.

Brian Long
Long time Delphi genius, Brian is also a dab hand with Borland's *Kylix*.



Andrew Channelle
Now studying 'culture' or some such nonsense, Andy still finds plenty time to write the news!

Maurice Kelly
Long time contributor Maurice is in the chair for HotPicks, selecting the best new software.

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 page 114

CONTENTS

The UK's best selling Linux magazine

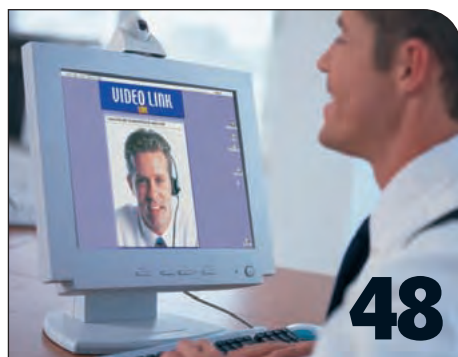
LINUXformat

LXF25 March 2002

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

Voice Over IP

Don't be left out when it comes to tele- or video-conferencing – and let Linux take the strain off your international telephone calls.



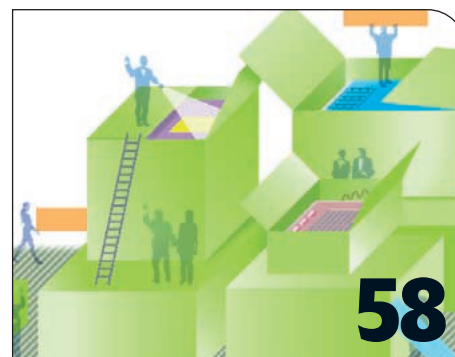
High availability

Network downtime costs a lot of money – unlike the preventative solutions available under Linux. Sleep more easily with five 9s availability.



What's DevFS?

In UNIX systems all devices are treated as files on the /dev directory. Find out about the latest way to interact with your hardware.



COVER FEATURE

TUTORIALS SPECIAL!

As we expand our tutorial section, a world of Linux knowledge awaits

Starting on page **67**

Save money and subscribe to Linux Format. See page 96 or phone 0870 444 8645

REVIEWS

24 OVERLAND LXN2000

Automate large backups with a robot in your library.



26 SUSE 7.3 PPC

Has the easiest-to-install PPC distribution kept its crown with yet another feature-packed release?

28 REDMOND LINUX

Read about the new newbie-friendly distro here and then try it out from the cover CD.

30 VOLUTION MESSAGING

Caldera are targeting SMEs with this easy to install and administer mail server solution – will it suit your business?

32 QUANTA GOLD

Web development made easy with theKompany's enhancements to *Quanta Plus*.

34 BOOKS

XHTML Essentials, *Programming Linux Games* and an interesting look at designing user interfaces.

35 CROSSOVER PLUGIN

View Shockwave and Quicktime websites from your Linux browser, using Windows plugins and *Crossover*.

TUTORIALS

68 EXT3

The journalled filing system with the safe and easy migration route. Bask in ext2 compatibility.

70 PARTITION IMAGE

Always be prepared for the worst. Don't delay – learn how to back up your partitions today.

74 PYTHON

An introduction to quick 'n' dirty object oriented programming.



78 CVS

Too many patches can break the code – unless you use *Concurrent Versioning System*.

82 PERL

Signals, sockets and pipes – writing a client/server program is simple with so many inter-process communications.

86 JAVA

Java programs are self-aware – or parametric polymorphism made simple (apparently).

88 KYLIX

Browse our coverdiscs with this natty little CSV reader. Learn shared event handling and several useful techniques.

95 SYSREQ

Knowledge of the magic key will get you out of all sorts of difficulties.



>> REGULARS

NEWSDESK	6
MAILSERVER	12
HELP WANTED	16
ANSWERS	18
HOTPICKS	42
USER GROUPS	108

>> FEATURES

ROUNDUP – CD BURNING	36
The best CD burning software on test	
EMULATORS – ACORN	62
Early Acorn1 to sophisticated RiscOS	
VISUALROUTE	94
Upgrade with our special offer!	

LINUXformat Coverdiscs



A DVD or 2 CDs packed with the latest Linux goodies **98**

VisualRoute and *Simply GNUstep*, with *Redmond Linux* on the second CD and several specialised distros on the DVD.

Please read the coverdisc instructions starting on page 98 before installing from the coverdiscs!



LINUXformat Newsdesk

Linux everywhere! – set-top home entertainment and iMac-alikes to Linux-only big iron and clusters; good and bad news for gamers; Linux for constabulary and DJs; and “just say no” to Word attachments.

It's all so entertaining

Moxi home invasion

This year's Consumer Electronic Show (CES) was the venue for the first product announcement from former WebTV supremo Steve Perlman. The Moxi Multimedia Centre (MC) is designed to be the heart of your connected home and combines five functions in one box: DVD and CD playback, MP3 jukebox, Personal Video Recorder (PVR), Cable/DSL router and allows you to browse the web, send instant messages and – the holy grail in the set-top box world – watch video on demand. Not bad for a machine that's supposed to be as easy to set up as a video. The device was judged Best of Show.

And that's not the end of the MoxiMC's talents. At its core is a custom Linux distribution which means with little effort the box can act as a hub for your home network as well as a link with the wider Internet (and it can be used as a firewall) and with the addition of separately purchased MCx remote terminals, you can add the same features to any TV or PC in your home using either connected or wireless networking.

While the interface is based almost exclusively on Macromedia's *Flash* technology, the device features some serious hardware including an upgradeable 80GB hard disk (good for 60 hours of video) and connections for just about

Moxi Media Centre – home entertainment with Linux



everything including firewire, USB and Ethernet or wireless 802.11.

“My entire career has been focused on delivering on the promise of home entertainment,” said Perlman, “At Moxi, I've assembled the ideal team, technology, customers and partners to make this a reality. We look forward to delivering consumers an integrated solution that simplifies and enhances their enjoyment of entertainment in the home.”

Get Real!

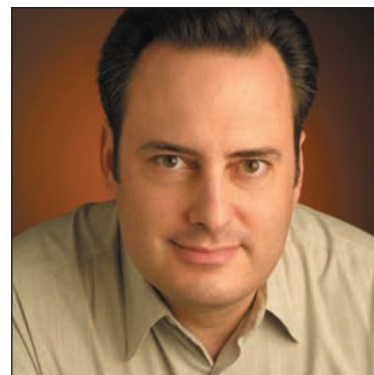
As well as the tie up with Macromedia, Moxi have also aligned themselves with Real Networks to integrate the company's *RealOne* player into the MoxiMC; a deal, says RealNetworks' Dan Sheeran that,

confirms the company's ambition to become the standard provider of streaming media. “Our relationship with Moxi will enable cable and satellite providers to deploy a new range of highly innovative set-top boxes giving consumers access to their favourite Internet media.”

Though the boxes should be available to buy before the end of the year, Moxi hope to make the biggest impact by convincing cable and satellite operators to standardise on the MC. However, Perlman won't have his technological lead for long, with similar devices coming from both Nokia and Pace.

Microsoft were expected to launch their rival Homestation at this year's show, but preferred instead to push the X-Box.

www.moxi.com/product.htm



Steve Perlman has been involved in many seminal computer projects including Atari, Apple and WebTV.

Desktops for the long arms of the law

'ello 'ello 'ello, what's all this then?

Britain's police forces could soon be using Linux in their endeavours to protect us. The Police Information Technology Organisation (PITO) have been charged with the task of assessing the feasibility of porting Plod's crime fighting software from Windows to Linux in a six-month study for

Project Valiant, to be undertaken by open source lobbyists Netproject.

Marco Dawson said the police were looking at Linux due to its industrial strength security and virus resistance. “There is a potential requirement for 60,000 desktop systems within the UK's police

forces, so we need a very good understanding of the deployment and migration issues involved.”

The Association of Chief Police Officers said no decision had yet been taken, and a Microsoft working group had also been convened.

<http://www.netproject.com/>



Linux game porter demise

Loki closes its doors

The world's biggest commercial producer of Linux games has closed its doors after a long fight for survival. "The idea with Loki was never to create a thriving Linux porting business. We wanted to create a Linux gaming industry," said Scott Draeker.

The news emerged in a leaked email to retailers suggesting that once Loki stock was gone, it was gone for good. Draeker later clarified the situation in an interview with Newsforge in which he said he was saddened by the closure, but would be looking forward to new challenges.

Over the past three years, Loki has been responsible for some of the most high-profile Linux game releases, but

critics have often argued that the games come too late and at an inflated price compared to Windows editions.

Loki, best known for porting games such as *SimCity 2000*, *Quake III* and *Tribes 2*, laid off the last of its staff at the end of January, after a 'sales collapse' over the holidays. A marked change from when the company filed for bankruptcy in late 2001, and Draeker was in positive spirits. "Based on monthly sales figures when we filed, we had every reason to think the reorganisation would be a success. Those monthly sales figures fell off dramatically over the holidays. With lower than expected revenues every month we were digging a hole." www.lokigames.com/

Coming soon: Neverwinter Nights

It's not all doom and gloom in the Linux gaming world though. A project we mentioned back in the mists of 2001 looks likely to come to fruition thanks to settlement of a dispute between developers BioWare and publisher Interplay. *Neverwinter Nights* will finally see a multi-format release (with Windows and Linux versions in the same box) 'very soon'.

Neverwinter Nights, you may remember, is an 'epic' roleplaying adventure game that can be played either as a single player or as part of an online team. However, its development and release has been delayed after a dispute erupted between the developer and the publisher. BioWare's Ray Muzyka said, "We're delighted that we have reached an amicable settlement agreement in our dispute with Interplay, and we look forward to completing the

development work on *Neverwinter Nights* and having Infogrames publish this product."

BioWare's previous successes include *MDK2* and the much-loved *Baldur's Gate II: Throne of Bhaal*. www.bioware.com/games/neverwinter_nights/



Neverwinter Nights – coming soon?

NEWSBYTES

According to the new release schedule, **KDE 3.0** should be available as you read this. The latest version is based around Qt 3.0.1 and comes with the usual complement of applications, utilities and games.

A survey by IT analyst **Gartner Inc.** suggests that Linux is running on servers in 15 per cent of companies in the Asia-Pacific region. And the trend appears to be growing. Gartner's report says Linux is particularly strong in countries such as Korea and China.

In Korea office productivity vendor **Hancom** are publicising an initiative by the government that will see HancomLinux Deluxe installed on 120,000 desktop systems. The package includes Linux OS and office suite.

Latin American governments are also getting the open source idea as the perfect way to prevent a technological divide opening up. The latest Brazilian project aimed at giving people access to the web could see the Ministry of Science and Technology subsidizing the sales of sub-\$200 Pentium systems running Linux and KDE. However, according to a report by **Britain's National Computer Centre**, the present government is failing to make the best of open source software which, the report says, could be 'of great benefit'. Dr Andrew Hopkirk, head of said: "There is clearly great potential for cost savings and increased efficiency."



Redmond Linux Corp have changed their name to Lycoris as "part of an ongoing effort to gain independence and brand recognition in the growing Linux software market". Their core product, Redmond Linux Personal has received a similar nominative makeover and becomes Desktop/LX. Prolific developer **theKompany.com** have announced the official release of *Rekall*, a "personal, programmable DBMS system for KDE". The product works with *xBase*, *MySQL* and *PostgreSQL* out of the box, and as it has a full set of widgets, can integrate with any KDE theme you happen to be using.

Hoyt Duff

The author is one of 800 Hoyts living in the USA and runs a little fishing pier when he's not dabbling with his computers.



COMMENT

Sellout or Sell Out?

Miguel de Icaza believes that basing **GNOME** on the .NET API puts he and his followers at an advantage when, it is assumed, .NET will sweep the Internet. The .NET concept is the Redmond Gang's vision of the inevitable future of the Internet.

What's happening here? Mr. de Icaza just signed off on backing away from the GPL for **GNOME** and now he is saying things like "They [Microsoft] have a beautiful security system..." and "Right now, we're compiling on Windows..." , as well as "...Microsoft really does develop some really interesting technology." Really?! By adopting the .NET API, de Icaza believes that he can cut development time and costs and more easily implement **MONO**, the open Source Version of Microsoft's C# – the framework of .NET, but could he lose more in the bargain? **GNOME** is, after all, the darling of the hacker crowd, being the desktop of choice for those who flaunt convention and decry the impurity of the commercial Qt widget set used in the more-popular-than-**GNOME KDE** desktop. As well, **GNOME** is the desktop of choice for the MS-loving (not) Sun who have decided to abandon the **CDE** desktop now used for Solaris. What self-respecting hacker is going to support a desktop that slips in bed so easily with the Anti-Linux?

What's the upside for de Icaza's **GNOME**? If .NET becomes the default definition of the New Internet, **GNOME** drags the Unix world with it into the Light, leaving non-.NET arch-rival **KDE** and other desktops as less-than-useful historical curiosities. Every Unix user becomes a **GNOME** convert.

What is the possible downside? Caught up as a vassal to the Redmond Gang, de Icaza and his **GNOME** are doomed to a Hell of marginalisation as Open Source chalks up a big loss to the Dark Side.

Either way, the "Sold Out" sign will hang on de Icaza's door. What is this guy thinking?

Big Blue's Linux big iron

IBM launch Linux-only bargain mainframe

Two new machines from IBM this month, the iSeries for small businesses and the more expensive and powerful zSeries, will be what IBM's Peter McCaffrey calls "The first pure Linux mainframes."

The projected cost of the average Linux-based iSeries computer would be about \$40,000 and using IBM's partitioning solution, can be split into 15 virtual Windows or Linux servers. The iSeries supports SuSE and Turbolinux distributions and features a special installation wizard for rapid deployment of new servers. The zSeries packs the same punch as "hundreds of servers" and would cost in the region of \$400,000 – half the average mainframe cost of \$750,000.

"These new Linux servers answer the call of every customer who is serious about reducing server sprawl and dramatically improving their total cost of ownership," said Bill Zeitler, eServer group executive. "Companies large and small are turning to 'virtual' Linux servers to save real dollars as they gain better control over their e-business infrastructure."

McCaffrey added that the new servers could take mainframes into whole new sectors. "It does allow us to reach different customer sets and different audiences that we couldn't reach with traditional full-blown mainframes." The use of Linux has also removed some of the complexities of server administration, making the whole experience more load-up and go.



Above: IBM's Bill Zeitler says companies can save real money with virtual Linux servers.

Left: The zSeries becomes Big Blue's first Linux only mainframe.

LinuxWorldroundup

As ever, this year's New York Exposition features a deluge of announcements from software and hardware suppliers of all sizes. We'll have a full report next issue, but here are the highlights

At the top end of the scale, Hewlett Packard's Chief Executive Carly Fiorina said she believed Linux exhibited the same pioneering spirit as the founders of HP and that the merger with Compaq would be good for the OS. In her keynote address, Fiorina said "We are more powerfully positioned to lead the march in open standards and Linux than either company could on its own." She also stressed the futility in concentrating on attacking Microsoft. "The reality is that Microsoft solutions on industry-standard hardware are a mainstay of the industry, especially on the desktop. The question for us is not will Linux dominate the world. The question is what part of the world will Linux dominate?"

Hewlett Packard also announced an alliance with Mandrake which will see the pair promoting Linux on HP computers.

Alcatel, Cisco Systems and Nokia have joined the Open Source Development Lab, becoming part of a focus group to

define what is needed for a 'carrier grade' server to service the telecoms industry.

IBM announced that its much trumpeted \$1 billion dollar investment is already showing dividends. Bill Zeitler said the company had recouped most of the outlay in "the first year in sales of software and systems." IBM also announced their first Linux only servers (see other story).

SGI workstations are being shunted off the desktops of animators at Dreamworks (Home of Spielberg and Shrek) in favour of Linux boxes. Dreamworks said the new system - used to create the upcoming all CGI film Spirit: Stallion of the Cimarron - cost about half what they paid for a similar suite to make Antz two years ago. The only gap in the Linux armour, says Dreamworks' Chief Technology Officer Ed Leonard, is a paint application - the company is said to have requested a Linux port of Photoshop from Adobe, but were turned down.

Registration & Travel

Show Attractions

Conference Program

My Show Planner

News Center

LinuxWorld 2002, NYC

Dr. Dobbs' **technet**
http://www.technet.com

CONFERENCES
January 29 - February 1, 2002

EXPO
January 30 - February 1, 2002

LOCATION
Jacob K. Javits Center, New York C

New York, New York – penguins in the big apple.

Server start-up Egenera used the show to announce their first customer: Credit Suisse First Boston (CSFB). Egenera formed to sell BladeFrame servers ranging in price from \$200,000 to \$1 million. CSFB were one of the original Beta testers for Egenera's servers, and their Linux kit is used for international routing, handling up to 20 million transactions per day. The servers use Power CockPit management software from TurboLinux.

Veritas have signalled their intention to release Foundation Suite - a system that allows one server to assume duties if a connected device fails - to Linux. Veritas previously sold backup solutions for Linux, but this is the first time Foundation Suite will be available for a non-Unix operating system. Meanwhile Veritas rival Legato announced a similar system for managing IBM's database software running on Linux.

NEWSBYTES

Mandrake have been celebrating a pretty good year which saw them grab the lion's share of the desktop Linux market. A series of polls carried out by users across the globe gave Mandrake between 35 and 45% of the market, with Red Hat and SuSE gaining silver and bronze positions. A spokesperson said the results "confirm that our strategy of Making Linux easy to use is meeting the needs of individuals as well as enterprises that require a powerful, yet easy to use operating system."



Mandrake have also allowed the first beta of Mandrake 8.2 to escape from The Cooker. The latest distro includes a few innovations including a pared down installation coming in at just 65MB, encrypted file system and a revamped Control Panel.

Last month we mentioned **Blue Linux's** refocusing on the education market. The team have put out their first release candidate complete with Adobe Acrobat Reader 4, Real Player 8, Flash 5 and fully anti-aliased KDE 2.2. Matt Jezorek said the distro provided a "fully open sourced operating system for schools and government."

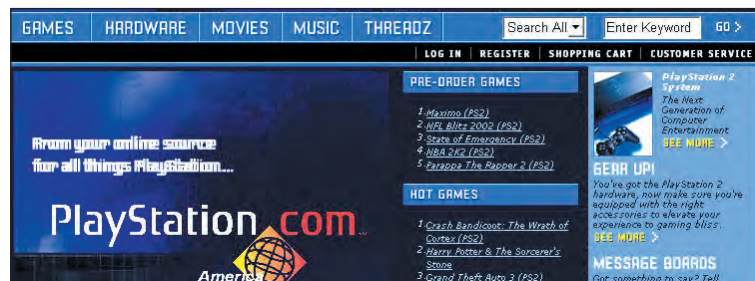


Bless your ears with the latest release from the OSS/3D project from **4Front Technologies**. Version 5.5.7 features, among other innovations, a 'Brainwave Relaxer' "based on psycho-acoustical models of how the brain perceives and interacts with sound to provide relaxation and meditation," the company says.

IDC have done a comparative study which concluded that Linux offered TCO savings of nearly 80 per cent over traditional Unix systems. To let the OS reach its full potential, enterprises should create a strategy for longer-term deployment, and migration from proprietary systems.

What's their game?

PS2 Linux US launch in May – but not UK?



After a successful launch in Japan Sony are to release Linux for Playstation2 in the United States at the end of May.

The upgrade kit is expected to cost \$199, providing users with a powerful Linux-based PCs for the bargain price of about \$400. The kit will be available initially through Sony's US website and will give hackers their first chance to get a look at the Playstation2 runtime

environment and system manuals.

Linux for Playstation2 was announced last June with a limited run of 2,000 selling out in Japan within a few hours. Sony went on to sell an addition 5,900 kits by the end of the year. The company has received nearly 10,000 requests for the upgrade from the US and in Europe over 28,000 people have expressed an interest.

There is a dedicated website for the project: www.playstation2-linux.com/

System specification

- Internal hard disk drive for PlayStation 2 (HDD) with 40 GB capacity
- Network Adaptor (Ethernet) – for PlayStation 2 – with 100 Base T Ethernet interface
- Linux Kernel version 2.2.1 (with USB device support)
- Linux for PlayStation 2 Version 1.0 software distribution on two DVDs
- gcc 2.95.2 and glibc 2.2.2 with VU assemblers
- XFree86 3.3.6 with PlayStation 2 GS support
- Computer monitor adaptor – for PlayStation 2 (with audio connectors)
- USB Keyboard and mouse – for PlayStation 2

Linux on the rumour mill

To buy or not to buy

The online Linux community went into apoplexy when a rumour spread the media giant AOL/Time Warner had set its sights on acquiring Red Hat.

The rumour first appeared at The Washington Post (which has a good track record on AOL scoops) and spread across the news sites with the speed and agility of an Outlook enabled virus. Opinions, as expected, weren't far behind. The article suggested: 'To counter Microsoft's desktop hegemony, New York-based AOL Time Warner could use the deal to couple its America Online software, the market leader with more than 33 million Internet subscribers, with Red Hat's OS technology.'

As the reactions to the proposals became a flood – including kernel hacker Alan Cox threatening to leave Red Hat if the rumours were true – AOL/Time

Warner spokesman Andrew Weinstein tried to set the record straight claiming the Washington Post story was simply not true. "AOL is not in negotiations with Red Hat." Investment analysts Merrill Lynch agreed, but suggested a deal of some sort was in the air: "We think there could be a strategic partnership/customer deal brewing," perhaps with embedded applications or the Netscape browser.

As the biggest Internet Service Provider in the world, an AOL 'desktop' would be another weapon in the company's increasingly hostile battle with Microsoft. In January AOL filed a private suit against MS claiming damages in relation to the Netscape browser, while the pair have also clashed on the provision of information such as stock quotes and sports news using Instant Messaging technology.

Jono Bacon

The founder of UK Linux, KDE developer and all-round nice guy, Jono Bacon is studying at Wolverhampton University.



COMMENT

Independent dependency

“In these days of vendors shaping and selling Linux in many different forms, we have seen some interesting collaboration from Free Software hackers and the commercial backers. The interesting thing is that many companies such as MandrakeSoft, SuSE, RedHat and others are all relying on the productivity of the Free Software world in order to sell a unit.

I feel that collaboration of this nature is something that was not only pre-empted by the rising popularity of Linux, but has also thrived on the mutual benefits to business and Free Software. We have seen substantial support from companies such as SuSE, IBM et al. Taking SuSE as an example, they have had Dave Jones working on giving Linux support for the latest processors – this has given Linux a place near the front in developing support for new architectures. This industry support for developing Free Software is an unusual step for the “it's my ball and you are not playing with it” business world.

The big question is regarding where to draw the line – how far can the business and Free Software world cooperate and manage conflicts? Luckily, for the Free Software world, our friends at FSF have developed the pretty robust GNU GPL which protects us from most problems. Although this licence is fine for covering the future source management and development freedom, it does not prevent certain paid hackers from bolting in advertising and links that make the product less generic – imagine seeing one company's logo when you use a piece of software in another distribution – this could lead to advert hell which has plagued the commercial operating systems world.

Once again, theory here cannot override practice; although there are risks available to the Free Software and business communities, the apparent advantages are bearing fruit and conflicts are minimal. Have we really seen the beginning of a true symbiotic relationship between these two differing communities?

Big up the Linux massive

Hear Linux get wicked

Advances in cutting edge sound and music control have never really been associated with Linux, but that's all changing with the release of *Final Scratch Pro*, a software and hardware combination that gives DJs the control they need with the added benefits of digital technology.

The system from Stanton Magnetics has been two years in the making and has received the seal of approval from techno luminary and *FSPro* beta tester Richie Hawtin. "When [fellow DJ] John Acquaviva and I first started messing around with it, it was a better concept than working apparatus. But they got back in touch with us later when it came together." The system is now so steady that they've done entire four hour sets without touching traditional vinyl. "Needless to say the benefits of such technology are endless. From being able to carry an infinite number of tracks without the added



weight of vinyl, *Final Scratch* sounds truly revolutionary."

Final Scratch Pro uses a special pair of 'records' and a digital I/O box to give the DJ total control over MP3 files stored on their laptop's hard disk, meaning an end to bulky record boxes and a limited number



of tunes (200 seven minute songs fit on a 2GB partition). The control is so fine that you cut, spin, scratch, and needle drop MP3's without losing the beat.

The system is imminent for both Linux and BeOS, with a Mac version in the pipeline. www.stantonmagnetics.com

Looks cool, sounds incredible. Can *Final Scratch Pro* finally kill vinyl?

Embedded Linux News

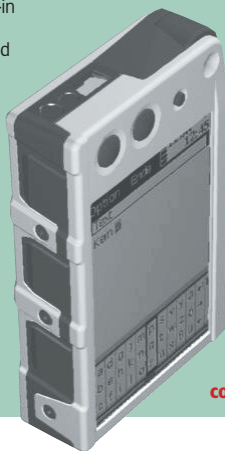


● You wait for one Linux PDA to arrive and then dozens (nearly) come along at once. First up is the Royal Lin@x from Royal Consumer Information Products. The device, which was demonstrated at this year's CES, is based around an advanced Intel 206MHz StrongARM processor with 16MB flash ROM and 32MB of RAM.

It also features 'outstanding colour graphics, built-in expandability, and the power and customisation of Linux'. Royal's Todd Althoff said: "The Lin@x provides immediate consumer value and delivers the features customers want in an affordable, open environment that can accommodate emerging technologies and applications."

<http://www.royal.com/>

● Also making waves is the rather thick (physically) Filewalker PDA from Germany's Invair – which at 3.4 x 2.2 x 0.74 inches is designed to be operated with one hand. This



one is based around the slightly slower Intel 133MHz StrongARM system-on-a-chip with 32MB of DRAM. However, instead of a colour screen, the Filewalker only manages 16 levels of grey, which should at least keep the price down. All of the included applications (the usual PDA stuff) are available in German, English, French, Italian, and Spanish.

<http://www.invair.de/FILEWALKER/filewalker.html>

● Version 3 of the Linux Terminal Server Project (LTSP) is available now. This thin client software "provides an alternative, extending the life of existing desktop PCs until they literally fall apart from old age."

<http://www.ltsp.org/>

● MontaVista's big news for the LinuxExpo was the release of MontaVista Linux 2.1, featuring increased hardware and software support and new tools. Based on the Linux kernel 2.4.17, it also offers MontaVista's real-time enhancements including preemptibility on the x86, MIPS, SH and PowerPC architectures.

<http://www.mvista.com/products/mvlf.html>

● MontaVista recently announced a further \$28 million dollars in funding, and an increase in revenues of 250 per cent.

<http://www.mvista.com/company/index.html>

Budget priced number crunchers

Eight way clusters on the cheap

The cost of building a Linux cluster just tumbled with the first product line from US vendor Rocketcalc: you can pick up the entry-level machine for just \$4,500. The basic Redstone-E is powered by eight Celeron 800Mhz processors and 2GB of system RAM, while the Redstone B (\$6,500) feature eight 1GHz Pentium III's and a massive 8GB system RAM. All of the computers also include a 100Mbps ethernet switch, devfs file system and a startup/shutdown time measured in seconds – thanks to the OS residing on non-volatile FlashRAM.

At the heart of the Redwood machines is a streamline embedded Linux distribution (Motor) built on the 2.4 kernel and tailored for cluster computing. Rocketcalc say one of the biggest benefits of Motor is that it is built on statically-linked binaries which allows 'integration of Rocketcalc



clusters with x86 Linux workstations by eliminating the possibility of library conflicts'. Any application that runs on your workstation is likely to run without modification on the cluster. Dr Bryan Lewis said Rocketcalc's intention was to provide high-performance, low-cost and easy-to-manage hardware and software to expand the application of cluster computing."

Save us from proprietary Word attachments

Stallman's Brave New World of openness

After helping to kill the concept of patents in web standards (perhaps), Richard Stallman has turned his ire to a phenomenon which shows no sign of letting up: MS *Word* attachments.

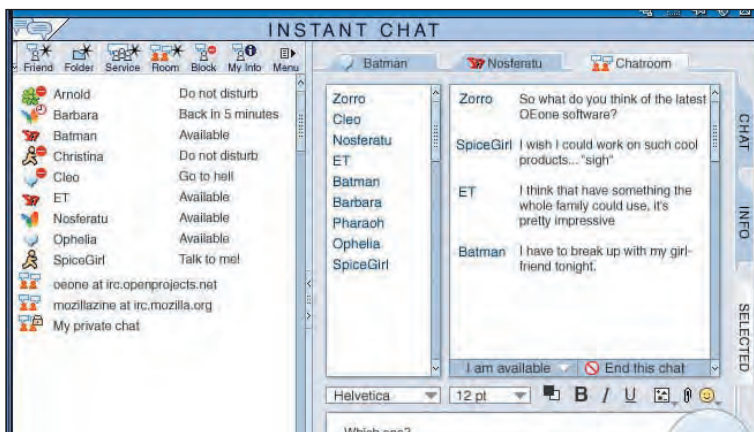
"Word attachments are annoying," Stallman wrote. "but worse than that, they impede people from switching to free software." He says a polite cease and desist request with people who send you *Word* files could be the start of a real force for change.

Although *Word* users should be pitied, Stallman claims the reliance on proprietary standards has an impact of users of Free Software too. "Someone I know was unable to apply for a job because resumes had to be *Word* files. Even governments sometimes impose *Word* format on the public, which is truly outrageous." Worse

than the effect on free software users though, the prevalence of *Word* documents may convince other users to stick with their proprietary applications. "They hesitate because they feel they must have *Word* available to read the *Word* files they receive. The practice of using the secret *Word* format for interchange impedes the growth of our community and the spread of freedom. While we notice the occasional annoyance of receiving a *Word* document, this steady and persistent harm to our community usually doesn't come to our attention. But it is happening all the time."

The solution, according to Stallman, is a concerted community-wide initiative sending back offending emails with a polite request to resend the information in "a non-secret" format. www.fsf.org

Linux PC as home appliance



OEone use Mozilla to great effect for the user interface.

OEone Linux iMac style thing

If you were attracted to the ease of use and styling of an iMac but put off by MacOS, the OEone Internet Computer could be the answer to your prayers. The OEone is based around an 800MHz PIII Processor and a 17" CRT monitor and retails at a competitive \$700. Unusually the computer foregoes the benefits of KDE or GNOME (though these can be installed – it is Linux after all) and instead uses a customised version of Mozilla 0.9.3.

As well as providing all the facilities you'd expect from a modern connected computer (TV viewing, MP3 playback, productivity) the Internet Computer also

comes with 100MB of online storage which can be sync'd with your local filesystem to provide either effective backups or access to your files from anywhere on earth. OEone's reliance on Mozilla makes a lot of sense when you realise the company's own developers did quite extensive work on the project's calendar and the Penzill framework and application suite.

The Internet Computer is the first of a planned range of products based around Linux, Mozilla and other open technologies. Find out more at www.ozone.com

NEWSBYTES

The Free Software Foundation (FSF) have announced they're now able to take orders and donations online. A statement read: "We were waiting for the RSA patent to expire. After RSA was released into the public domain, we were waiting for someone to set up a server for us. Paul Fisher finally did! See www.fsf.org for more. Norway's Opera continue to release 'technology previews' of the Opera 6 web browser. So far, these releases have proved to be pretty stable, fast and compliant."

The Chinese government recently awarded contracts to supply operating system, productivity and anti-virus software to six of Chinese vendors, including Red Flag. The deals are an attempt to stamp out copyright violation that is said to be rife in municipal government – a condition of the country's acceptance into the World Trade Organisation. Awarding contracts to 'indigenous companies' is, said analysts Gartner, an effort to support local economies at the expense of outside interests – however, some users pointed to the furor that erupted when Taiwanese programmers inserted anti-communist messages into the first Chinese edition of Windows 95



Redflag – I always said Linux was a communist plot.

Efforts to get Linux onto the 32-bit Psion organisers have progressed steadily for years at <http://linux-7110.sourceforge.net/>. Now the Psion 5MX is up and running with a GUI – *picoGUI*. Even the touchscreen works. <http://www.picogui.org/> Next step is making X work: <http://thomas.de-ruijter.cx/projects/psion/> Mozilla have announced their 0.9.8 Milestone – improved Address Book functionality, page setup (for printing), MNG/JNG support, native-style widgets on winXP and OS X, dynamic theme switching, improved BiDi support, speed, stability and footprint improvements.

David Cartwright

David Cartwright is an IT consultant who specialises in providing Linux systems and solutions.



COMMENT

A standard for goalposts

I've written before about how wonderful it is that commercial software vendors are becoming more and more enthusiastic about producing Linux versions of their applications. The main problem with the combination of Linux and commercial applications has, however, been that the applications often don't keep up with the Linuxes.

If you write a program for Windows 2000 (say) it's easy to match the standards. There is only one Windows 2000, and the standard interfaces don't change. With Linux, however, you have a number of different vendors, whose opinion of what should and shouldn't be in Linux (and what kernel is the right one) varies – so how do application manufacturers keep up? It's bad enough having to cater for the differences between (say) Mandrake and RedHat, without keeping up with new versions of each too. I have at least two Linux machines running old versions because the commercial packages that I'm running on them won't work on the latest release.

Hurrah, then, for the Linux Standard Base (LSB). This is an attempt by pretty well all the important parties in the Linux trade (including Linux distribution vendors like Mandrake and Red Hat and application vendors like Oracle) to agree on a standard Linux platform so that the application vendors can spend more of their time writing the code and less of the time worrying about what weird effects they'll get because Linux vendor X uses (say) a newer version of glibc.

The LSB defines all of the key properties of a "standard" Linux kernel, which means that the software writers know where the goalposts are. Not only do they know where the goalposts are, but the LSB reduces the number of goalposts to one, where you previously had Mandrake goalposts, Caldera goalposts, RedHat goalposts, and so on. And although there will be developments in the LSB from time to time, the idea is that you should be able to change your code just once, not once for each distribution.



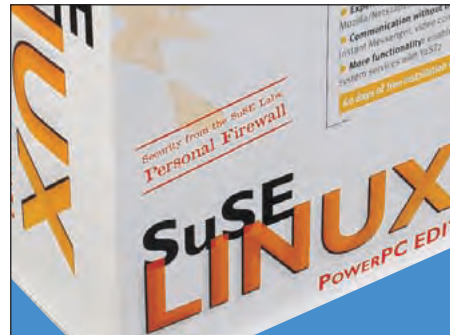
Mailserver

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

Where's SuSE?

I have been a subscriber to your magazine since the first issue and think that it is great. There are however a couple of things that I would like to see. Firstly, I think that you should include a full copy of SuSE Linux on the cover disk, not just the live evaluation version. I know that SuSE Linux is a huge product and couldn't fit on on CD, so you could include the Live Eval version on the CD and the full DVD version on the DVD cover disk. I currently use Mandrake Linux and haven't used SuSE for quite some time, but I would like to use it again (the full thing, not an evaluation as I want to run it as a web, mail, smb and LDAP server).

The next thing I'd like to see are specific to the tutorials section. Now that the 2.4.x version of the kernel is in wide circulation, some of the tutorials should be aimed at the new features that are available. The main one I would like to see is one for setting up a SOHO network using the new firewalling features. I know that it is possible to set this up very easily in Mandrake



SuSE not for covermounting.

using the *Control Center* (which is great), but if other readers are anything like me, they would like to know how to do it manually so that they have a better understanding of the techniques used.

The other thing that I'd like to see in the tutorials is something on LDAP. LDAP is a great tool, but a lot of people don't know what they can do with it, never mind how to use it. I think

that you should have a tutorial on what benefits LDAP brings over older technologies and how to implement it.

I know that it takes quite some time to write tutorials, so I won't expect to see them any time soon, but it would be good to see them in the future.

Thanks for providing us all with a well valued publication.

Colin Anderson, *via email*

Thanks for your feedback. The SuSE issue is a tricky one, because they don't actually want their current distro to appear on coverdisks, only the evaluation version. Their thoughts are that if you want to run SuSE, you should try one of their boxed sets, which seems reasonable enough.

Thanks for the suggestions for tutorials. Home networking is something we should definitely be looking at in the near future – though I hope you spotted our mini-router reviews, which offer a sensible and reasonable-priced option for home users.

★ Letter of the month

This month's winner receives a copy of **XHTML Essentials** by Michael Sauers and R. Allen Wyke

New beginnings

I'm not a beginner with Linux, and I've been using it a while now, but I remember the days when I was really stuck and I often asked myself 'Why did they make this darned thing so hard?'. Then I would always realise my mistake and everything would just click.

However, if complete newbie Linux users had some sort of solid, easy to follow, right from the beginning tutorial on all things that are necessary to first timers (e.g. installing software and hardware requirements), then this would surely be of a great help, and stop so many people who I have seen from dipping their toes in the Linux pool and pulling them sharply back out again.

I thought your tutorial in the Xmas issue of LXF entitled 'Introduction to Linux' was a

wonderful guide and very helpful to people in this predicament. I myself like to help out others, especially when it comes to Linux, and therefore I have (over several months) developed my site - The Ultimate Linux Newbie Guide.

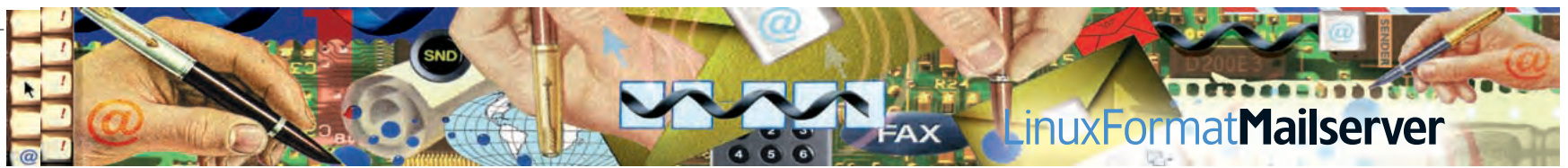
In ten chapters, I've covered the essentials (criticisms on this point are welcome), and I've popped things in like a forum, which has turned out to be quite busy. I write to you at *Linux Format*, to help spread the word of this site, so that we can continue the 'Introduction to Linux' and help our adventurous newbies out! This site is completely not for profit as I believe in keeping the open source idea very much a reality. I do hope this will benefit LXF's readers, especially those who are new, or have never even used Linux.



You can visit my site at www.linuxnewbieguide.com

Best Wishes,
Alistair J Ross, Unix System Administrator,
XBOLT Network *via email*

Thanks for your comments on our own guide, which we will be carrying on our disks in future so that all readers can benefit from it. I've had a brief look at your site and thinks it's pretty impressive – I hope you get lots of feedback. In the meantime to expand your brain a bit further, we'll be sending you *XHTML Essentials* by Michael Sauers and R. Allen Wyke (Wiley, ISBN:0471417645).



Install on ME

I recently purchased a DVD copy of your magazine and I must confess it, and Linux in general, has left me puzzled!. Because I successfully installed and used Linux on a number of systems four years ago I thought I would have a go using it at home. I started by trying to install TurboLinux off a relatively recent *PC Plus* magazine – it would install, but then Windows Millennium ignored it and would fail to offer it as a boot option.

I decided it would be wise to look for something that was bang up to date – just in case. I saw Libranet on your DVD and decided to try that instead. The problem was the DVD wouldn't boot and the install instructions relating to this situation referred to folders and files that don't exist on the DVD. I tried creating an installation CD but this got me nowhere as the CD had no bootable files on it.

I was considering a subscription but it seems no one in the Linux world has heard of Windows Millennium when writing installation routines, and not enough time and trouble has been spent on making instructions clear and relevant to the DVD. I've looked in detail at the magazine and the DVD for answers and now I'm stuck.

Brian Lawrenson, *via email*

Windows doesn't recognise any version of Linux, or indeed any other operating system other than those also from Microsoft. In order to boot into Linux you either need to create a boot floppy, or install a bootloader such as *LILO* or *GRUB*: the installation routines for distros will give you the option to install either or both of these. You could also use commercial bootloader software such as *OS Selector* or *System Commander*.

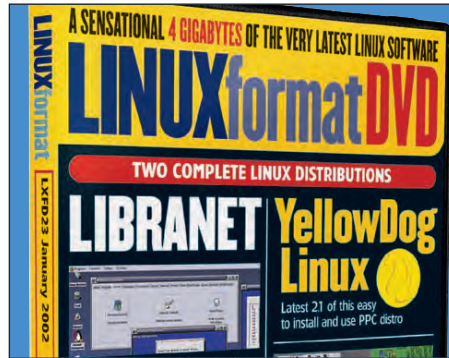
The Libranet version on our recent CD was in the form of an ISO image: the easiest way to install from this is to burn the CD and boot from it. The image is bootable.

Debian please...

I would be very grateful if you would consider putting the Debian distro on your cover cd sometime soon. Your magazine has featured many Redhat based distro in the past on it's cover CD or DVD but rarely has there been a copy of a Debian-based distro (apart from Debian for the Amiga). Considering that Debian is favoured by many linux users, that it would be very convenient for them to have a CD set at hand. It would also give some users the opportunity to try a distro that isn't developed along commercial lines like the majority of others.

David Craig, *via email*

Debian 2.2 was included on the DVD for *LXF* 24. We have never discriminated against Debian, but as stable releases are further between, it doesn't



The DVD contains ISO images to burn (to disk).

feature as regularly on our disks as some of the other distros.

Who on Earth?

In response to your Jan 2002 article "What on Earth is Co-location" I would like to mention another provider in the UK which you did not mention, and which has provided myself with excellent service for the past 2 years. For £99 per month you can either co-locate or rent a RAQ4 or RedHat Linux machine at www.dedicated-servers.co.uk, they offer good control of the Linux machines through the use of *Webmin* as standard for *Apache*, *MySQL*, *Bind* etc., or you can manage via *Telnet/SSH*, and support is very good. Last year their standard linux server specification changed for the better and they upgraded (on request) my machine for free when I renewed my contract, server reboots are guaranteed to be done within 30 minutes when requested, I tested and they do stick to their word ;-). I can highly recommend Dedicated Servers.

<http://www.dedicated-servers.co.uk/>

Kind Regards,
Anthony Ogden,
via email

Thanks for the recommendation.

Upgrade?

Hmmm, after all that hype about Windows XP I thought I'll give it a go (bad investment, I think).

First of all the installation program overwrote my dual boot without any warning. (Why? – I am asking myself). Then XP complained about my memory configuration. I have a 256 MB card and a 128 MB card, which work fine under SuSE 7.2 and even Win98. Then XP generated some sort of an installation report with some unexpected errors: my Canon scanner, my Kodak camera and – believe it or not – my Epson printer are *not* supported by XP, however all that gear works fine under SuSE 7.2 and Win98.

I thought if you upgrade an Operating System you do it to make a step forward, not five steps back. The memory configuration might be an excuse — but that is what Microsoft never learned to do that other OSs could do years ago. But a standard USB device? I don't think that this is unacceptable.

Well, did I get my money back? No. Once the Microsoft box is open ... Did I get any help? No. They think if it's not the memory then it must be the processor (I certainly will not change the processor).

Well, XP is still not running on my (one year old) system but SuSE is and I shall not be bothered to use any MS application anymore.

Stefan Korn *via email*

Poor support for existing hardware seems a common problem for those who have contacted us after installing XP. Microsoft's position is that only tested drivers get shipped with XP, but as most manufacturers aren't going to go to the trouble and expense of reworking drivers for old hardware to meet MS standards, it seems likely that many older devices are unlikely to be supported.

Eagle eyes

I have read/subscribed to *LXF* on and off since the early issues and I have enjoyed it thoroughly (and still do).

Whilst reading Mailserver in the January issue I saw Mel McWeeney's letter (headed 'Taxman'). I seem to recall reading in one of the early issues that a reader had an identical problem to Mel's – he didn't want to buy a laptop containing Windows software. The reader then read the Licence Agreements that accompanied the software and discovered that if he 'flattened' the hard drive and returned the OEM software and

Manufacturers and resellers should start getting the message quickly. Probably. Alternatively they will decide that Linux users are troublemakers.

books to the shop, he would be entitled to a refund.

If this is still valid, I thought it would be well worth letting other readers know about it. At the moment, preloaded Linux is scarce because the demand is low. I believe that this low demand is artificial because it isn't challenged. If Linux users started returning the Windows software and demanding a refund because it is unused and the packaging still intact, manufacturers and resellers should start getting the message quickly. Probably. Alternatively, they will decide that all Linux users are troublemakers and ban





◀ them from the shop. Or something.

Anyway, keep up the good work and all of that.

Ray Bayley, Dartford, Kent

It seems to be a matter for the individual shops. Some will sell you a Windows-less machine with some badgering.

You've been told

I think that there may be a kernel(?) of truth to the notion that some linux users/advocates do in fact secretly like the notion of exclusivity etc. Back in the days of RH 4.2 I went from Windows to not knowing my ass. It was a shock and a humbling experience. Advice to Barry Snelson:

- 1) Think about exactly what you want your machine(s) to do.
- 2) Make it so.
- 3) Leave it alone.
- 4) Upgrade only when you have to.
- 5) ...and possibly get another machine to experiment with, that way if it doesn't work you can leave it for a while.

More than happy to leave the bleeding edge to others (most of the time),
Ernie Hirose San Jose, California

You didn't mention *OpenPKG*, developed by Cable & Wireless Deutschland GmbH. To quote from the official (www.OpenPKG.org) site:

"Currently Solaris, Linux and FreeBSD are officially and fully supported. Additionally, unofficially and/or partly supported are NetBSD, OpenBSD, and Compaq Tru64."

If this system works well on all Linux flavours, I think it could be what Matt needs.

K.Rhodes, Paris (France)

But how many people are actually supporting it?

Codebreaking

I recently purchased your December 2001 issue and as I have been interested in cryptography for some years it was with much interest that I turned to the encryption article on page 30.

I would like to dispute the statements made in this article that I consider incorrect or only partially correct.

Page 30. "- the first use of computers in cryptography was actually in an attempt to break the Nazi Enigma code of World War II."

The Enigma code was broken before World War II by a Polish National and his work was passed onto the allies just prior to the outbreak of hostilities.

While Ronald Rivest, Adi Shamir and Leonard Adleman have been attributed with the discovery of the algorithm of RSA public key cryptography, it was originally conceived by James Ellis in

1969/70 and Clifford Cocks developed the algorithm in September of 1973. In 1974 Malcolm Williamson, in an effort to prove his friend's algorithm wrong, proved it was correct and in the process developed what has become known as the Diffie-Hellman-Merkle key exchange.

My source for this information comes from "The CODE BOOK" by Simon Singh (ISBN 1-85702-879-1). This book is a very



See *The Code Book* for more on cryptography.

readable history of the development and use of cryptography.

The above is not intended to be a criticism of either the article or the magazine but merely an effort to enlighten you on the development of cryptography in the latter half of the 20th Century.

Being a long time Slackware user means that I am not particularly interested in the modern distributions but I am able to obtain gems of information from your magazine

Keep up the good work in bringing Linux to the world.

John Briggs, via email

While most of the media's attention is on the shocking state of some or other public service, copyright and patent law seem to get little or no public outing.

Thanks for your rather vivid recollections. I hope most of our readers remain at least on speaking terms with their anatomy, but the point is well made.

Packages

In LXF22 (Christmas 2001) you published a Linux Wish List sent in by Matt Darcy. Number one on his list was for a "universal package for Linux".

Thanks for the additional info, and I'll certainly look out for the book. I think nevertheless that computers were employed on the Enigma code. I can't pretend to be an expert on it, but I do know that the Enigma machines went through several variations and it was possibly to crack new codes that computers were employed.

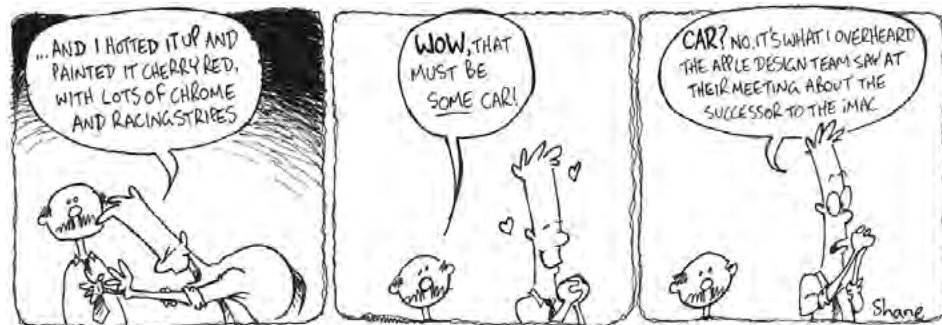
Praise for Drummond

Good magazine and plenty of good stuff on the DVD. Firstly I would like to commend Richard Drummond for his article on 'Get to grips with Fonts'. I thought it was informative, clear and most importantly it was not condescending to ordinary Linux users like myself. I find that a lot of articles, docs and information on linux are unclear; long winded; full of jargon and very patronising in attitude. I want and seek information that is clear, relevant and honest even if this information has to be graded for the novice, the intermediate user and the expert. If

Helpdex

By Shane Collinge

shane_collinge@yahoo.com





I wanted the other I may as well stick with Windows.

Could your magazine do an article, or could you point me in the right direction for good information, on setting up email, using POP3 and an ISP on a standalone laptop using *KMail* and *Mutt*? I have had terrible bother with this and have not succeeded yet. I am now using RedHat 7.3 and still *KMail* can receive and send emails but when I try to send a large attachment, over 50 KB, the program stalls. I have recently, well since your last magazine, been trying *Mutt* 1.3.23i and can receive emails but I can not send emails over the internet only to my local mailbox. I get the returned message with 'error the internet provider can't recognise the address'.

Are there problems with either program sending large attachments? I know that *KMail* has had problems with this having written to their forum and got told by a reader not to send large attachment.

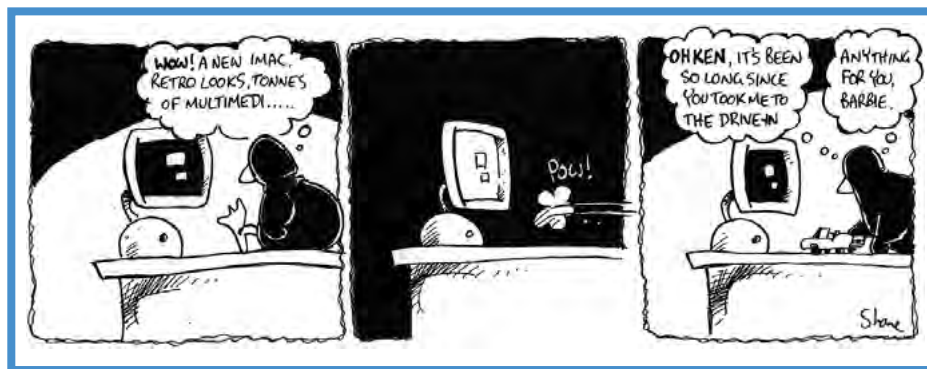
These are two separate issues but I like *KMail* and I also would like to give *Mutt* a shot because it seems very fast – so maybe a separate article. Jim Quigley, *Buncrana, Donegal*

I won't mention it to Richard, he'll only get big headed. As to your mail problem, it seems that *Mutt* is not properly configured for the outward mailserver.

KMail doesn't have a size restriction (I'm not sure which version you are using but 1.3.7 seems fine), but your ISP may do – you should be able to discover this on their own technical help site.



DMCA – an offensive, four-lettered word?



Indirect action

Having been a reader for some time I would like to congratulate you on a useful and entertaining magazine. I am particularly impressed in your ability to maintain such broad coverage. I think there is a real danger of reaching a condition whereby the coverage of each section is so dilute as to be almost useless. You seem to have avoided this pitfall.

Having followed the debate about the Digital Millennium Copyright Act, your article in *Issue 22* spurred me into action so I wrote a letter to my MP, Kenneth Clarke. I had a very speedy reply in which he said he would raise the matter with the appropriate Minister and let me know the response.

I really would urge other readers to contact their MPs over DMCA type legislation. A lengthy letter is not required; it will be a bit off putting. On the whole MPs would rather receive an "Executive Brief". So just one paragraph, short and to the point would suffice.

The DMCA and the whole area of Encryption must be watched very carefully at the moment as it is possible that legislation could be enacted which will have very far reaching consequences indeed.

Kind regards,
Simon Heaton, *via email*

I'm glad to hear it. Unfortunately while most of the media's attention is on the shocking state of some or other public service, things like copyright and patent law – which can often effect the individual more than they realise – seem to get little or no public outing. Do let us know what Ken Clarke says.

Express to nowhere

I thought you may be interested in the following correspondence I have had with VMware, especially as your latest cover disk includes VMware Express, a product I would be interested in using. I have been unable to get a license code for VMware Express as they have stopped selling it.

Clearly the VMware workstation pricing is



VMWare – leaving the budget sector.

prohibitive for recreational/ household use: Their strategy is obviously focussed on the corporate sector.

Any chance of an evaluation version of *Win4Lin* on a future cover CD?

As an aside, I have succeeded to get *Workstation 3.0* for a virtual Win95 session up on my humble Cyrix PR200MMX with 64MB RAM. It was slow to install but once up and running is quite responsive using *Office 95*. Sadly, after two late night sessions I still can't get the networking to work.

Martin Knott, *Edinburgh*

Yes, it seems that after their experiment, VMWare are discontinuing their entry level product. We are unsure of the specific reasons for this, but it does rather make *Win4Lin* a more viable alternative. We'll be sure to flag it up appropriately if we do carry a demo. What demos would people like to see on the CD? **LXF**

Mailserver Hot Topics

We have introduced Mailserver Hot Topics to help gauge your opinions on the things that matter most. Please feel free to continue writing in on any subject you like (except the glues we use on the CDs, everyone's heard enough about that), but we would be extra keen to hear your views on the hot topic of the moment. Without further ado, next month's topic is: **Viruses and Linux - are we afraid or are we prepared?**

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 question – 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: linuxformat@futurenet.co.uk

Help wanted

From preserving freedoms to helping online gamers be aggressive and competitive, **Richard Smedley** discovers all sorts of ways to get involved in the Free Software community.

Our first request for help is aimed at anyone using email – we can't get much more inclusive than that amongst our readers.

As you may have read in this month's *Newsdesk*, Richard Stallman (The founder of the Free Software movement, widely known as RMS) has decided to make a stand against the insidious spread of proprietary software through the need to open and read Microsoft *Word* attachments.

Most of us have got used to converting these to readable form one way or another, only to find that the document is a simple piece of text that could have been pasted into the body of the email. For the *nix desktop user they are just one of life's little annoyances (though to those who still get their email through a shell account

they are a much greater inconvenience).

However RMS points out that the very prevalence of such attached formats reinforces their use, as *Word* attachments become a *de facto* standard and are almost mandatory for most job applications.

To get involved simply send back emails sent with proprietary attachments to the originator. Write with a polite but firm request for information in an open format, accessible by all. For sample replies and the reasoning behind the campaign, see:

<http://www.fsf.org/philosophy/no-word-attachments.html>

Should you wish to escape the cares of the world and its incomprehensible attachments you may have indulged in a little

recreational violence through the world of online gaming. *Quake III Arena*, *Half Life Team Fortress* and *Return to Castle Wolfenstein* are all within the reach of Linux users. What have they got to do with Open Source programming? Well the games ISP, Jolt, hosts leagues for all these games (and some others, which are apparently playable upon another platform). Stephen Smithstone has been busy developing a league management system, but now needs a hand. If you know PHP, MySQL and HTML now's your chance to polish these highly marketable skills.

Contact Mr Smithstone at partylinux@btpopenworld.com www.jolt.co.uk

Lastly those of you who have managed the horrific intricacies of *Sendmail* configuration are invited to contact Matthew Carson to help "save a large high school in the north west from having to run *Exchange Server*."

Tips needed

Linux-tips.net is looking for Experienced Linux users to come to their site and share knowledge with the newbies. They are aiming to construct a site where new people can come and get a Tip of the day, or just browse the archive for something they did not know.

As co-founder Tom says: "We started this project in November and recognize the potential for a huge Linux tips archive. We really need the experienced Linux users to come a give them a hand. Tips, ideas, scripts, and useful new software."

<http://Linux-tips.net/>

In return if you need help getting Symantec i-gear running in a mixed Linux/Windows environment, then Mr Carson is your man. matthew_a_carson@hotmail.com

NMS

Bug-free CGI scripts

Although advocates are always saying that you can do anything, in any way, with Perl, one of its strengths is in CGI scripts. NMS consists of a set of CGI scripts that are intended as drop-in replacements for the scripts at Matt's Script Archive (MSA).

MSA is a repository of CGI scripts written in Perl by a programmer called Matt Wright – probably the most popular repository of CGI scripts currently available on the internet. Matt Wright wrote the scripts in 1995, and they became extremely well known and popular. However, as Wright says, "I wrote that code years ago when I was just learning to program, but have lost the time and motivation to keep it up." Despite bugs and security flaws the scripts continue in use, so last year the London Perl Mongers decided to write a series of drop-in replacements for Matt's scripts. "This means that anyone who uses one of Matt's scripts should be able to get the nms replacement and

just drop it in in place of the original script. Everything should then work exactly as before, except that the user will sleep easier knowing that their web site is that little more secure," says Dave Cross, compulsive Perl coder.

The target audience may well know very little about Perl, so the scripts will run under Perl 5.004_04 (or later); will *not* use any non-standard Perl modules – the target audience is not expected to know how to install modules from CPAN; and they must run with no errors or warnings under **use strict** and **-wT** – this is particularly important to teach good habits to those who learn coding Perl by examining these scripts.

"Code Review – We don't claim to be infallible, but with enough other people looking at our code we can find all of the bugs and insecurities.

Testers – We need people to install and try out the nms scripts. If you have problems with the installation, or in running them, please let us know.



The London Perl mongers – partying and coding is what they do best.

Documentation – Our target audience are not Perl experts. We need clear and easy to follow documentation. Currently the documentation could be a lot better.

Developer: – We can always use more developers. If you know Perl then we'd love to have your help. Want to get involved? <http://sourceforge.net/projects/nms-cgi/>

The EFF

Freedom on the electronic frontier

A Quick FAQ, by Martin Linklater.

Q Who the hell are the EFF?

A The Electronic Frontier Foundation are best summed up by a slogan printed on one of their stickers: 'Protecting Rights in the Digital Age.' In a recent article, the LA Times described the EFF as the 'American civil liberties union for nerds'.

Launched in 1990 by a group of Internet pioneers concerned about how governments were beginning to police the internet, the EFF is dedicated to making sure civil liberties are protected in the technological frontier. Seen by many as an organisation which wants to make life easier for crackers and copyright pirates, the EFF vehemently defends the rights of free speech, privacy and anonymity on the internet. Above all, the EFF is making sure that the rights and freedoms you hold in the real world, will be upheld in the new.

Q But they are American – that's nothing to do with me.

A We are the 51st state – like it or lump it. Although America is distant in geographical terms, the UK is profoundly effected by how the USA regulates the internet. And without the internet, this thing called Linux

would never be running on your computer. Defending the right to play DVDs on Linux – the EFF is involved. Defending the right to highlight weaknesses in encryption – the EFF is involved. Concerned that the recent terrorist attacks are being used as an excuse to remove fundamental freedoms – the EFF is involved. The EFF is very concerned about your future freedoms – even if you don't realise it.

Q Why bother?

A Although Free Software is moving ahead by leaps and bounds, there are many challenges which need to be addresses with the help of legal representation. Whether it's defending the rights of a Princeton Professor who was stopped from presenting a cryptography research paper at a USENIX conference, or fighting for the freedom of 2600 magazine to publish hyperlinks on their website, people can sometimes need superior legal representation – the EFF provides this. Bad laws don't fix themselves – people who oppose the laws fix them. We may all need the EFF's help sometime in the future – give them your support today.

Oh, and they are a registered



Cindy Cohn, EFF Legal Director – success in the Bernstein crypto case.

charity by the way; so don't worry about the 'excess profits'. ;)

Q OK, so how can I help?

A Become a member, make a donation. Running a legal firm is not cheap – even one created as a charity. Donate as much as you can spare – as much as you value your freedom. To quote 2600 magazine: "The

internet, once the shining beacon of free speech, cultural exchange, and open expression is fast becoming the exclusive property of big business and oppressive regimes. At least, this is how it appears in their minds. We cannot let our own perceptions be corrupted by this invalid premise."

<http://www.eff.org>

Sharpen your quills

End complaints of poor documentation in Open Source projects

Just to prove that you don't need to code to help a project I did a quick trawl for those needing nothing more than a few words putting together for the documentation.

We know you can read (you're reading this) so if you can also string two words together why not spend an hour or two helping to document a project that you like.

Most of the people using a piece of software will not have had a hand in its creation, as such they are dependent upon documentation for getting around the program, and for getting the most out of it.

As Max Albert of *Web Admin for Half-Life* (WAHL) says in his appeal for document writers "Most docs will be for end users. They don't need to be too technical but rather explain how to use the features. You can probably figure this out by playing with WAHL.

There will also be a docs that will have instructions for installation. This one will need to be very specific. It will

include steps for setting up files and permissions in a *nix, and setting up a database and tables in MySQL."

WAHL is a tool written in PHP and MySQL. It is intended to give game admins the ability to administer a *Half-Life* server over the web. WAHL is built to work with *Half-Life* mods and support *AdminMod*. As well as writers WAHL needs lots of other help – see the page at:

<http://sourceforge.net/projects/wahl/>

ButterflyCP aims to be the graphical user interface for all popular open source server technologies. With the laudable goal of solidifying server management, from email and domain setup to database management and software updates. The project is at an early stage, however the only thing that the developers claim to be lacking is a document writer.

Have a look at the project and if you are (or can become) familiar with

how the GUI works, contact them now. <http://sourceforge.net/projects/butterflycp/>

NES DevKit is a development kit for the Nintendo Entertainment System (NES) for both Windows and Linux – built with *Delphi/Kylix*. The purpose of this dev kit is to make the programming of games and demos for the NES a lot faster and easier. They aim to make "the best DevKit for the NES!"

As well as document writers, the project is looking for coders, so if you would like to practice your *Delphi* (and your C++) take a look at: <http://sourceforge.net/projects/nestdevkit>

Of course some documents are more technical than others – take Computer Aided Software Engineering (CASE) tools, for instance.

vmtlinux aims to be a "free, easy to use, quality, open source CASE tool for the Linux platform, that supports code generation and reverse engineering,

and Distributed Concurrent Designing."

They need help with the quality manual (i.e. document and coding standards); project proposal and project plan – as well as a document on task allocations.

<http://sourceforge.net/projects/vmtlinux/>

For more on the advantages of CASE tools, take a look at:

<http://www.linuxgazette.com/issue67/gilliam.html>

Your help wanted

If YOU have a project that's in need of anything from artists and beta-testers to web-designers and, er, something beginning with Z, we want to hear about it. Email us now at linuxformat@futurenet.co.uk and give us some details of your project, and what sort of help you are looking for. Please include plenty of info about the project!

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!

>> Experts this month

Whatever your question is, we can find an expert to answer it – from installation and modem woes 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.



Richard Drummond is an experienced programmer who can answer queries on a variety of subjects. A keen Debian user, he's also our resident Java guru.



Nick Veitch is the editor of the magazine, and answers your easy questions! Or indeed anything to do with *Grub*, *LILO*, *netatalk*, vi...

Drive space

Q I am installing Linux on a 8GB hard disk, the problem is that Linux only sees 1.9GB.

I know that this hard disk is working because I have tried it in Windows and DOS *FDISK*.

The motherboard does not need upgrading because it's new, and it see the rest of my hard disks (two 40GB). I have tried Red Hat 7.2 and Mandrake 8.1. I get the same problem with both of them.

Thanks alot

Malkit Singh

A The first thing to check is the BIOS, and see what the BIOS thinks the drive is. Check that the drive geometry matches up with that of the drive itself, otherwise the BIOS will tell Linux that the drive is a different size from what it really is.

If DOS *FDISK* can 'see' the whole drive, then the distributions you are using may be only offering you the free space on the drive to partition. RedHat and Mandrake use different methods in order to partition your drives, but they will both allow you to delete any Windows partitions, then create new Linux partitions in their place.

Network stew

Q As you could guess, I need some help more than I would like to admit! Before the questions: thanks for a solid source of info (just sucking up... ;)

Probably you will have some good answers and suggestions to point me in the right direction. I am running Mandrake 8.1 (download) on several machines, so that I can experiment and learn: dual Celeron, DVD, Xpert2000, with Ensonic sound card, the other machine is a K6-2, with a RagePro II 3D, CDROM and CD-RW, and a SB Live. None quite state of the art, sufficient to keep me interested and busy...

After some (almost) 3 years of using Linux, I do consider myself still new to this. Here are my problems:

1. No more super mount (out-of-the-installation) in this distro. Any easy way to enable it – without recompiling anything, or any major "surgery" for that matter? I'm also very lazy ... :)
2. Is there any way to make visible the content

```
david@niamh:~ (pts/4)
# fdisk -l
# This is an automounter map and it has the following format
# key [ =mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

boot          -fstype=ext2                :/dev/hda1
Removable     -fstype=ext2,sync,nodev,nosuid :/dev/hdd
cd            -fstype=iso9660,ro,sync,nodev,nosuid :/dev/hdc
floppy        -fstype=auto,sync,nodev,nosuid  :/dev/fd0

/etc/auto.misc: 9 lines, 482 characters.
```

autofs is a kernel-level auto mounting service, which is easy to configure.

of the CDROM of one machine to the others on my intranet, so that I could install RPMs, play DVDs or music directly from one machine to the other? With NFS working, I cannot read the CD from the other machine.

3. The most pressing — my internet setup doesn't allow me to read my email with *KMail*. First the set-up: for ISP reasons, less exposure, experimental, etc., I have a win98 machine running (a very stable and good) *Proxy+* (www.proxyplus.cz) in front of a 486 running Smoothwall. The 486 is linked to a Pentium 166, running Mandrake 8.1 (no GUI) "hub" to which the other machines are linked. The only way *Fetchmail* was able to see my linuxfreemail account was through *Transconnect*. Fetchmail correctly reported the content of the account, closing with an error after the attempt to transfer the first message, stating it was unable to transfer any of the messages. The *Proxy+* provides ftp, http/https, real audio proxies, socks ports, POP3 and smtp mail server, etc. Any suggestions on how should I best configure email access on such a set up?

So far I can listen to TV and radio stations, and with the exception of the email and Flash enabled sites, all works. This leads to the next question:

4. Flash is installed (!?) as a default in this distro – may not be the latest incarnation – I've never seen it work under *Netscape* or *Konqueror*, and when I hit sites requiring Flash, I get some 5 to 8 *Konqueror* windows opening faster than I can close them. Very annoying. The "locating, downloading and installing" of the correct Linux

version for this distro is another annoyance. I seem unable to get the right one, and most of the time some deps are not satisfied. Any statically linked RPMs?

On a happier note: *Ogle* (0.7.5 and 0.8.2 — just updated few days ago!) works fine (the cli — as I'm still fighting the gui!), so DVDs are a go on the dual Celeron system! And the 486 DX2-66 started with only 8 MB of ram (runing also *KDE* — very sloooooow — from the *Idiot's guide* ..., a Caldera 1.3 distro ... :) ...) and a 120 Mb HD (now it has 16 Mb ram). Since Feb 1999 it was stopped 5 times (except for another extra 3 times for upgrades!): 3 for vacation and twice because of blackouts — no complaints here!

Thanks for your support, and hope to see more interesting stuff about Mosix and MosixView in your pages.

Have a great New Year,

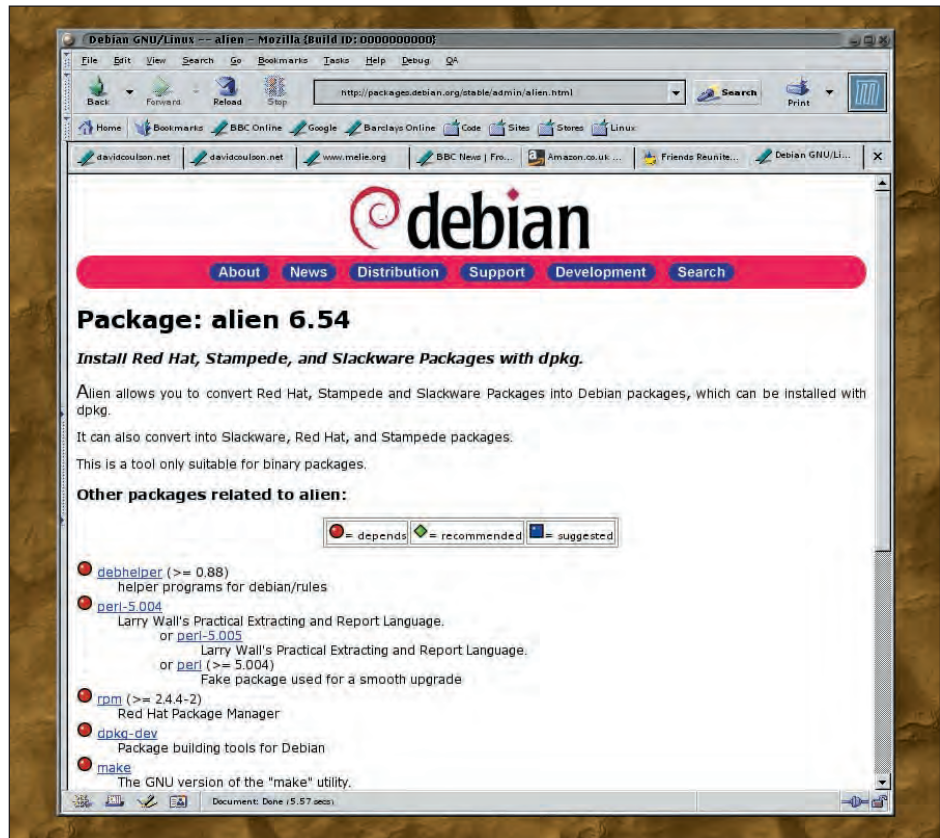
Marius.

A To respond to your questions in order;

1. Super mount is not really used anymore. 2.4 kernels, and newer 2.2 kernels, have a system known as *automount*, which consists of a kernel module and a user-space daemon to control the mounting of devices. Modular distributions should have the *autofs.o* module, which you can just **modprobe autofs**. Many distributions will have the *autofs.d* package available, or you will be able to download it from their site. Setting up *autofs* is quite straight forward, as you only have to edit */etc/auto.master* and, generally, */etc/auto.misc*. The defaults should give you sufficient guidance as to how to configure it properly to suit your machine.
2. Sharing CDROM content is best done over NFS, although since you have to mount the CDROM in order to do this, then you need to do something a little different in order to have it work properly. NFS only functions with mounted file systems, so as you can't mount an audio CD, or mount a DVD such that it will work properly. You can, however, use *cdfs* (<http://www.elis.rug.ac.be/~ronsse/cdfs/>) to mount your CDROM drive, which shows ISOs, CD-audio, and other information as a file in the file system. This means that you can mount the CDROM using *cdfs* locally, then share that directory. You can then either play audio tracks on a remote machine, or



Just get Smoothwall to do the dial-up, IP masquerading and to make the tea.



Alien allows you to install packages from different distributions.

mount a shared ISO image locally, or remotely, using the loopback file system. This doesn't work with DVDs, at least at the moment, so your only choice for playing DVDs over a network is to **dd** the entire DVD image from the DVD, then either NFS share it over your network, or otherwise copy it to the machine which you want to play the DVD on. As these are generally in the region of 5GB, you may want to reconsider putting a huge image file on a machine just to play a DVD.

3. Is there a need for the Win98 machine? Smoothwall is a perfectly adequate dial-up server, and as it can use IP masquerading, it is completely transparent. Proxying POP3, smtp, and so forth through an extra machine is somewhat unnecessary, and it really doesn't sound as if it's doing everything you want it to. To remove the Win98 machine from the equation will take a little network reconfiguration, but all services will work happily through it.

4. Flash is purely a number binaries which you put into your *~/netscape/plugins* directory. You can check to see if Flash is installed properly by looking at 'about:plugins' in *Netscape*. *Konqueror* can also use *Netscape* plugins, so you will need to check on the documentation for it to see exactly where it wants you to put them. There should be no need to use RPMs for Flash, as it's just a couple of files, which you can find on *macromedia.com* as a tar ball.

Having a statically linked RPM would be rather pointless, as you would end up with huge binaries, and it would obviously make the whole reason behind having a dependency based system void. As

long as you only download RPMs from the distribution's site or FTP server, or look on *rpmfind.net*, and ensure that you get the correct RPM for your distribution, you should have very few problems with dependencies, and if you do, you just need to download the specific RPM which fills that dependency.

Alien binary

Q I'm running Debian testing/unstable and I bought a Lexmark Z13 printer and am having problems getting it working. Its a USB printer so I tried to set it up using the USB guide in *LXF#22*, but this refused to work (There were no devices listed under */proc/bus/usb/*). So I decided to try the linux drivers that Lexmark indicated on their web site. I downloaded these and followed the directions. Running **sh lexmarkz13-1.0-4.sh** started the installation procedure. It listed the printer port as */dev/usb/lp0*. Clicking next started the install, but then the following error appeared:

error: cannot open Packages index using db3 - No such file or directory (2) error: cannot open Packages database in */var/lib/rpm* while executing "exec rpm -i --test \$rpmfilename"

and the rest of the install fails. I think the problem is that I am not running a RPM based distribution. Any idea how I can get round this problem?

Thanks
Nick Wilson





Linux's USB sub-system can support a number of different Ethernet chipsets, but you need to know which one to use, find a list of drivers at <http://www.linux-usb.org/>

A You are quite right. Debian uses *dpkg*, where as *RPM* is used on distributions such as RedHat and Mandrake. If you are able to extract the RPM, you can use the *alien* program to convert it to a *dpkg* package, and then install it. You can get *alien* with:

```
apt-get install alien
```

It is worth noting that Debian 'testing' and 'unstable' are two different distributions of Debian. 'testing' is what will become Debian 3.0, or Woody, and 'unstable' contains unstable packages. If you have both testing and unstable listed in your `/etc/apt/sources.list`, then it will use the packages from unstable as they will have higher version numbers, which may be a problem if the packages do not install properly.

USB networking

Q I have ordered via mail order Belkin's USB Connect, which will enable me to network my laptop and my desktop PC under Windows. Both my laptop and my desktop are set up as dual boot systems, running SuSE 7.3. Would it be possible for me to network my laptop and desktop under Linux via the USB connection? If so, how would I go about it?

I would appreciate your help in respect of this matter.

Yours faithfully,

John Wallis

A Linux supports a number of USB based Ethernet NICs, but unless you know which chip set the NIC has, it's difficult to say if it will work or not. You may need to upgrade your kernel to the latest of the 2.4 series, so that you are sure that you have the most recent versions of the modules and that you know that anything which may be available is not missing. If you still have problems, www.linux-usb.org lists all of the Linux USB drivers, so there may be links to Ethernet modules which are not included as part of the vanilla kernel tree.

Losing the source

Q I'm relatively new to Linux (about 1 Yr) and it's been co-existing with my Win2k setup. Being a keen newbie I like to try software, when I can get it to work, and end up downloading lots of interesting things. I am getting confused as to where to put downloads etc, especially the .tar.gz and .rpm files. I see a lot in the magazine about using `"/usr/src"` or something in the `"/usr"` directory. Coming from a Micro\$oft background, I tend to just put stuff in my home directory and install from there. I get the impression that this is maybe not the best thing to do. As ever with Linux, there seems to be no hard and fast rule regarding this. I just wondered if the LXF team had any thoughts. I like to keep a tidy machine as far as possible and would welcome any advice.

Thanks for the time,
Dan Marks

A Where you put downloads is not an issue - It's what you do with them afterwards.

Packages, such as RPMs, can be installed from anywhere, then you can either delete the RPM, or archive it away for reference. You will probably just download everything into your home directory, then either install them straight from there, or put them somewhere else with other RPMs, then install.

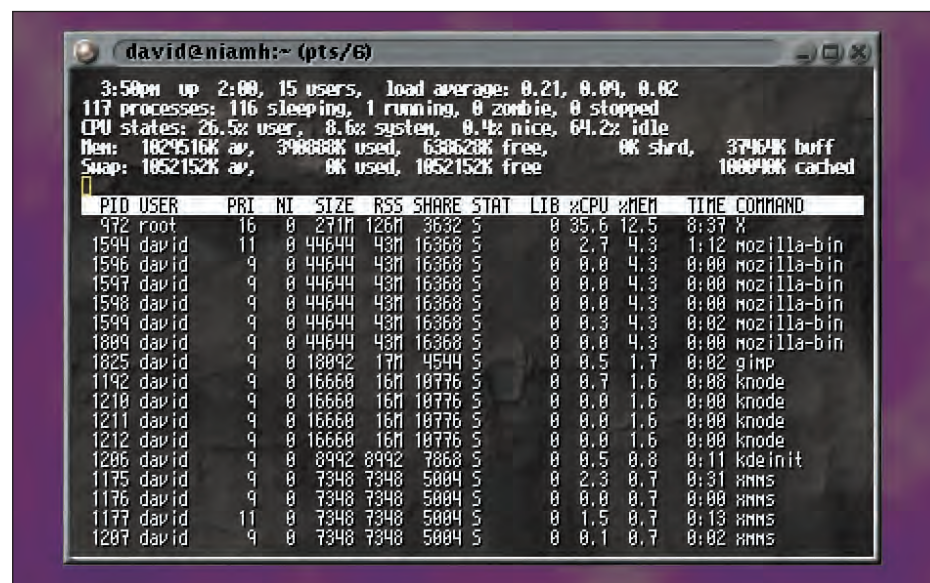
Tarballs can, again, be installed from anywhere. People sometimes use `/usr/src` to store source code, but it really makes little difference at all. As long as you perform the **make install** as root, assuming that you need root access in order to install the binaries, you can configure and build the source tree from anywhere you like. Very rarely does anyone keep source code hanging around, so you're not going to end up with a directory containing everything you've built for the last six months.

Memory hog

Q First I'll follow the example of many of your readers and tell you folks that you are doing a good job. I subscribe to another UK-based Linux mag and although I also enjoy it and learn from it they just don't quite have the same quality as you guys.

Anyway, here's why I am really writing: I'm enjoying getting away from MS and am almost happy using *StarOffice*. However, when I tried it recently on a 200MHz machine with 32MB of RAM and Mandrake 8.0 *StarOffice* took over 10 minutes to open. I was hoping to use Linux on that machine and begin to convert other in our office to Linux, but no way. I think perhaps I was being a bit too ambitious.

On my own machine (Toshiba laptop, 600MHz, 128MB RAM, Mandrake 8.0) *StarOffice* runs okay, but when I use *top* or *ps* view processes it shows that *StarOffice* is running multiple times/processes and using lots of memory



As with *StarOffice*, Mozilla uses multiple threads, but they share memory.

Missing drive cylinders?

Q I recently installed two 20GB drives into my Linux box. I have created the partitions and formatted the drives. The drives show 100% free but the space available is 17.4GB instead of the 18.4GB.

Brett Brock

A The first line of inquiry would be to use **fdisk -l /dev/hdX**, where 'X' is the drive letter. This will list all the partitions on the drive, along with the drives C/H/S geometry. You

will want to ensure that all the partitions on the drives are concurrent, in that the end cylinder for one is followed by the start cylinder for another. You may also want to check that the final partition has an end cylinder equal to the number of cylinders of the entire drive. There will always be space lost on the partition, due to superblock storage, and reserved blocks, but this should not have such a great impact on the amount of storage space available.

```
david@niamh:~ (pts/6)
niamh 3:49pm Sun Jan 27
david:~$ sudo /sbin/fdisk -l /dev/sda

Disk /dev/sda: 255 heads, 63 sectors, 4462 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           13      104391    fd  Linux raid autodetect
/dev/sda2                14          4462    35736592+    5  Extended
/dev/sda5                14          2251    17976703+    fd  Linux raid autodetect
/dev/sda6             2252          3781    12289693+    fd  Linux raid autodetect
/dev/sda7             3782          4291     4096543+    fd  Linux raid autodetect
/dev/sda8             4292          4422     1052226    fd  Linux raid autodetect
/dev/sda9             4423          4442     160618+    fd  Linux raid autodetect
/dev/sda10            4443          4462     160618+    fd  Linux raid autodetect
niamh 3:49pm Sun Jan 27
david:~$
```

fdisk can show you your drive's partition information, so you can see if there is any free space.

(see output of **ps** below):

PID	USER	RSS	%MEM	COMMAND
5370	dave	1040	0.8	/bin/sh /usr/bin/opera
5371	dave	14360	11.3	/usr/lib/opera/5.05_tp1/opera-static
5444	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5461	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5462	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5463	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5464	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5465	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5466	dave	45776	36.1	/usr/share/office52/program/soffice.bin
5685	dave	1104	0.8	top
5802	dave	724	0.5	ps -aopid,user,rss,%mem,command

Is there something wrong with my installation of *StarOffice* or is this normal?

Thanks,

Dave (Dar es Salaam, Tanzania)

A *StarOffice* is a multi-threaded application, so it spawns a number of processes to handle different tasks. These processes can share

memory between themselves, so all of your *soffice.bin* instances are using the same 45776bytes of memory, or around 45MB. *StarOffice* is not a small application, and does require a fair amount of memory to run properly. 32MB of RAM is not near enough for *StarOffice*, so you can either think about upgrading the memory of the system, or look at an alternative, such as *KOffice*, in order to use a little less memory and get the same things done. You may also find other small X applications which do things similar to that of individual parts of *StarOffice*, so *freshmeat.net* is a good place to look.

No boot

Q I run a dual boot Win98 and Win2K system, and want to install Mandrake 8.1 on another partition. I have fully prepared my drive, backed up, and all the rest, and prepared my three disks from the DVD — at least I thought I had.

The problem appears to be in writing the bootable disk. I suspect that the root of the problem is in the section that asks which emulation is required e.g. FDD, hard drive, no emulation (I use *Nero 5*). Whatever the cause, I get a disk that starts up, but cannot continue onto the setup screen.


Without guidance on this I will not be able to use any distros from the DVDs, and I'm probably not alone. Could we please have a magazine article and, preferably, a page on the web site

with a step by step walk through of converting the ISO images on the DVD to usable CDs? Meanwhile, if you could give me a quick guidance, I would be grateful.

C. Anne Wilson

A As you've not given any indication as to what error you see when you try to boot from a CD, it's difficult to isolate the problem. Your first option may be to try to write the CD from a different package, and see if you experience the same problem. The Eltorito file on the CD is almost always a floppy disk image, identical to a boot floppy. CDs either boot, or they don't — there is no emulation, or anything like that involved. What happens after that is entirely up to the boot image.

As you've already burnt the three CDs, you could just build a boot floppy which contains the Linux kernel image required to install the system, and boot using that. You will find the image on the Mandrake web site, or within the ISO image when you mount it through the loopback interface.

As the ISO images on the DVD are exactly the same as those downloaded from the Mandrake site, so any problems you continue to have with the images will hopefully be addressed there. Mandrake have a number of mailing lists for their users, so you may want to check archives, or post a message of your own, to see if anyone else has experienced a similar problem. 

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 X to work" doesn't really mean anything to us if we don't know things like what version of X 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 and answer all questions. If we don't answer yours specifically, you'll probably find we've answered one just like it. We can't really give personal replies to all your questions.

WRITE TO US AT:

Linux Format, Future Publishing, 30 Monmouth Street, Bath BA1 2BW or email: lxformat@futurenet.co.uk

LINUXformat 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: Does it have a competitive price?

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.



5-4 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.



THIS MONTH...

Overland Neo LXN2000 >>

Data security comes at a price — but convenience and cool extras are included free. Will you stump up for this tape library and its speedy robot? **p24**



<< SuSE 7.3 PPC

From the iMac to the RS/6000, will PowerPC computers benefit from SuSE's latest upgrade? We give the good news and the bad **p26**

Redmond Linux

Another Seattle-based company enter the OS business with claims of user-friendliness. However this one may be on the right track **p28**

Quanta >>

theKompany claims their product means "web development made easy." Find out if this useful Qt-based tool lives up to the hype **p32**



Volution

Need a mailserver for your company, but don't have a full-time sysadmin to deal with the headaches? Try Caldera's solution **p30**



<< Books

Transition to XHTML or take a deeper look at the design of user interfaces. Also this month, Loki Software present a useful games programming tome **p34**

Crossover

Codeweavers new WINE-based app allows the use of all those annoying proprietary browser plug-ins under Linux, you can even read Powerpoint files **p35**



BACKUP LIBRARY DEVICE

Overland Neo LXN2000

■ **MANUFACTURER** Overland Data Inc. ■ **WEB** www.overlanddata.com ■ **PRICE** From £16,000

Nick Veitch looks around a library which contains no books, but could be the answer to your backup dreams.

The Overland Neo LXN2000 is compatible with SDLT and DLT8000 drives, and the Seagate LTO Ultrium 200 drive. The unit we tested was fitted with dual DLT drives. The drives are, naturally, hot pluggable, so if you need to remove one for maintenance, cycle your drives or just add a new mechanism, you can do so without having to take the library

off-line. SCSI Ports at the back of each drive connect to the library controller, which will then connect to your host controller. LVD and HVD SCSI options are available, as well as fibre channel should you need the throughput.

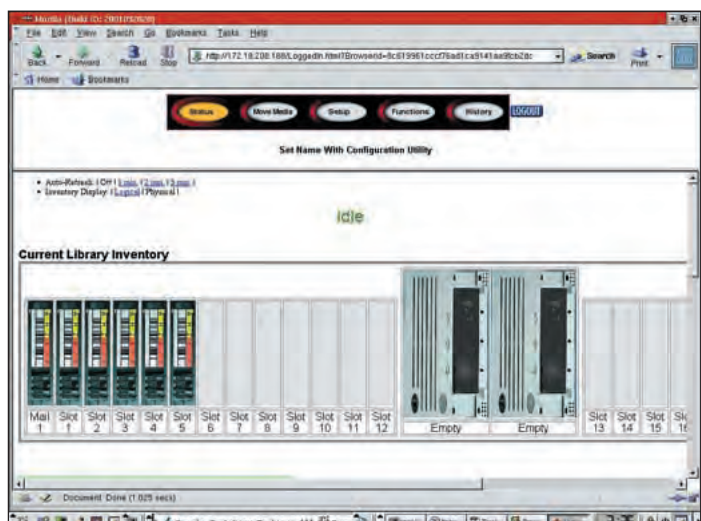
There are two variants of the Overland LXN2000 – desktop and rackmount, and both come in a very

sturdy, nicely styled case. Some amount of care has been taken to ensure that the library *looks* cool, which is apparently very important these days (The unit can power on just by touching the control panel, which obviously ups the cool factor). Two electronically controlled doors either side allow access to the tape magazines, while the centre of the

panel is occupied by the touch screen control panel. You will need to, at the least, set up the networking parameters for the unit here, though the rest of the admin functions can be achieved remotely through the web interface, of which, more later.

There is also a small viewing window above the control panel, through which you can catch occasional glimpses of the robot as he goes about his duties.

The DLT magazines can hold 13 tapes each, giving a total storage capacity of just over 1TB (native) depending on the type of tapes used.



The status page shows a photorealistic diagram of where the tapes are.



Pretty much everything can be set through the WebTLC interface.

Arkeia

You need good backup software to get the most from this library device



A flexible system of drivepacks and savepacks gets the most out of a library device.

The Overland unit can be accessed using a variety of backup solutions, and even directly. We tested this unit with Knox's *Arkeia* software (although you could equally use *NetVault* or other commercial packages).

The *Arkeia* software has a specific module for dealing with the Overland LNX2000 series, which may require separate licencing from your main

Arkeia software, depending on the version you have.

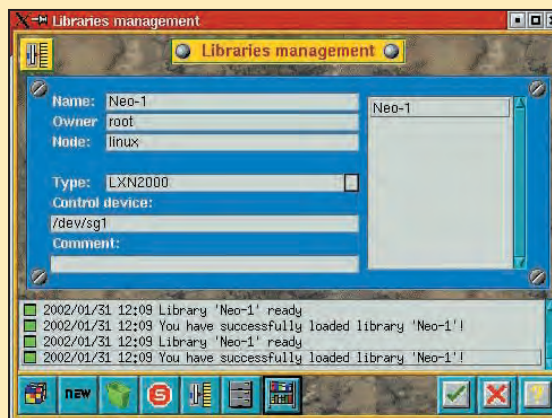
Arkeia provides a networked backup solution. A central server will be physically connected to the backup device, and clients on the system can use this to backup data (although the client can be run on the server too).

Arkeia is designed to run either from the command line (which is useful for

***Arkeia*'s LNX2000 module provides support for driving the library, and tracking the tapes in it.**

backing up by scripts or *cron* tasks) or through a graphical client. The graphical client is based on a neutral X environment, so it should run on pretty much any X enabled setup.

The LNX2000 module for *Arkeia* can handle all the features of the library device, though you will need to use the web interface or control panel to set specific library options.



The primary magazine features a mail slot, so you can insert or remove tapes without taking the library offline – very useful if you need to perform a one off backup without disrupting the routines of your regular backup cycle.

While the unit contains no backup server of its own, it does run a server of sorts, which will allow you to remotely check on the status of the device, and manage many of the options you would find on the main control panel, including shuffling the tapes about. This embedded server, called the *WebTLC* interface, obviously requires a network connection (an ethernet 10/100 port is included), and the server can be set to a static IP or

use DHCP as required.

At the very least this offers an alternative method of finding out the status of the library if there is some problem with any of your scheduled backups – although your backup software should report any errors, this way you can get a second opinion!

WebTLC will also allow you to fetch system trace information from the unit for diagnostic purposes, should your problems require more investigation. Several types of trace information are available, and can easily be converted into a human readable form.

The *WebTLC* interface also allows firmware updates to the onboard Flash ROM, either by fetching them from

remote ftp sites or directly uploading files from another client.

In use, the robot can usually manage to load up a tape in under ten seconds, so your backup times aren't going to be significantly effected by waiting for tapes to load. Rebooting the library can take a minute or so, as the robot then has to check the location of tapes. The tapes themselves can be barcoded for easier management. A barcode reader is included on the robot, but only Overland barcodes are supported – if you need more, you will need to order additional labels from Overland.

Of course, the Neo LNX2000 can be used as a separate module, or part of a bank of modules, allowing you to scale your backup needs. For extra

guarantees of reliability, or increased throughput or storage needs, multiple modules can be deployed. Up to eight modules can be combined into a single library, giving greater capacity and throughput.

Although some aspects of the touch screen control panel were a little confusing (it wasn't always clear under which heading the function you were searching for might be found), it certainly provided all the controls you could possibly want. Our efforts to confuse the library by simulating all sorts of faults, were to no avail. The stated MTBF for the unit is over 250,000 hours (excluding the drives, fans and power supply), which was obviously beyond the scope of this review to test, and a Mean Swap Between Failure of 2 million cycles (so you should be able to backup and restore over 70 Petabytes of data before you run into trouble).

It's hard to fault this unit on it's performance. Apart from a little more clarity in the controls, and the option for IP based backup (which is under development), the Neo currently incorporates the best technology in an easy to use, reliable, scalable unit.

The nearest competitor in terms of specification is probably Exabyte's 430 series, with a smaller total capacity and less user friendly on-board controls, though at a lower price. If the features are worth it to you, the Overland is the only choice. **LXF**

What is a library device?

A library device such as the Overland unit reviewed here is designed to make managing large volumes of backup data easier. The basic components are a number of tape drives, a magazine to hold tapes, a robot to put the tapes in and out of the drive and a controller card, which manages the tapes and the robot.

The library device allows you to create backup strategies using a number of individual tapes, and makes

managing of those tapes easier and almost completely automatic. Without a library, backing up data greater than your tape size would involve someone having to manually change the tapes in the drive – probably not something that you'd consider if you are running continuous backups on a largely unattended server. The library device can store many tapes, and manage multiple tape drives too to allow more flexible backup scenarios.

Linux Format VERDICT

Ease of use	9/10
Features	9/10
Performance	10/10
Value for money	6/10

Expensive, but currently sporting the best range of features and technologies.

Linux Format RATING

8/10



LINUX DISTRIBUTION FOR PPC HARDWARE

SuSE 7.3 PowerPC Edition

■ **DEVELOPER** SuSE AG ■ **WEB** www.suse.de/uk/ ■ **PRICE** £49

Wake Up, Little SuSE! A bit more testing would be a good thing. It doesn't all run smoothly for Bill Von Hagen.

Aside from the occasional clever remark about SuSE Linux being the world's most frequently released Linux distribution, SuSE usually gets nothing from me except respect. Unlike many Linux distributions, SuSE invests directly in the future of Linux by making substantial contributions to Linux projects such as the *ReiserFS* distributed filesystem and the *KDE* desktop. SuSE also actively supports Linux on both the X86 and PPC platforms, and sells Linux as a business solution for specialized uses as a mail server, firewall, and so on. Even the frequency with which they release their Linux distributions is essentially a feature, showing their commitment to making sure that SuSE Linux devotees have the most powerful, up-to-date version of Linux possible.

All of this praise and positive commentary makes it even harder to understand why the PPC version of SuSE 7.3 seems to have been released without all of the quality control, testing, and general attention to detail that is usually the hallmark of a SuSE release. I had problems upgrading or installing SuSE 7.3 on two out of three of my test machines, which were a Lime iBook, and two G4 towers, one with the older *Yikes!* motherboard and the other with the *Sawtooth* motherboard.

Upgrading from 7.2

Both my Lime iBook and *Yikes!* G4 tower were already dual-booting Mac OS X and SuSE 7.2, so my first test was to simply try upgrading the existing installations. On both of these machines, SuSE's graphical installer (YAST2 — *Yet Another System Tool 2*) failed to deal with the existing installations. The iBook was running SuSE 7.1 with all of the recommended patches from SuSE's web site, which YAST2 took to mean the system wasn't running a well-known version of SuSE that it knew how to upgrade. After trying this a few times, the answer was clear — reinstall from scratch. This worked fine on the iBook,



except that it never recognised my Airport card, and would only access the built-in ethernet port.

Attempting the upgrade on my *Yikes!* G4 was much more straightforward — it didn't work. It wouldn't recognise that Linux was installed on my second disk at all. This essentially ruled out an upgrade, but an interesting point was that I couldn't subsequently install from scratch on

that disk, because the text version of YAST (more about that later) detected partitions on the disk (created when installing SuSE 7.1), which it claimed it couldn't deal with. No problem — I figured that I would just boot into the Mac OS, initialise the disk, create the partitions as specified in the excellent SuSE docs, and then start from scratch. Unfortunately, my Macs all use Mac OS X as their Mac OS of choice, and the OS X drive formatter doesn't support the creation of Linux Swap partitions. You can get

around this by simply creating an unformatted partition, which SuSE assumes should be used for swap. However, it would be really nice if the SuSE docs discussed disk formatting with Mac OS X — after all, if we're interested in running cutting edge Linux, it isn't too much of a reach to assume that people will be running the cutting edge Mac OS, too.

Install from scratch

Starting with a clean slate is always easier than upgrading, but the installer for SuSE 7.3 still displayed a few problems. One of the scripts used during the installation process ("checkproc") couldn't be found by the installer, which was disconcerting rather than fatal. Aside from this, doing a clean install on my iBook and *Sawtooth* G4 went as smoothly as one could hope.

Unfortunately, on my older G4 with the original *Yikes!* motherboard, the installation process was like a holiday in Dante's inferno. Not only did the YAST2 installer alternate between hanging and crashing during each of my install attempts (though always in different locations), the YAST text-mode installer also exhibited the same problems. Suspecting that perhaps my machine had too much memory or a flaky hard drive, I pulled SIMMs out of the machine and even replaced the auxiliary hard drive, all to no avail.

SuSE does a great job of detecting available hardware and configuring the kernel and the X Window System appropriately in most cases, my previous observations notwithstanding. This sometimes has wider implications than simply selecting a specific set of drivers as a loadable kernel module. For example, the graphics chip and screen in my Lime iBook was only fully supported in *XFree86* v3.3.6 rather than the more powerful *v4.1.0* that is also included with SuSE 7.3. Similarly, the default settings for the ViewSonic monitor on my newer G4 were somewhat conservative, but the monitor itself was detected correctly and could easily be pushed to the higher resolution demanded by my endless quest for screen real estate.

SuSE also shows their commitment to Linux in items such as the

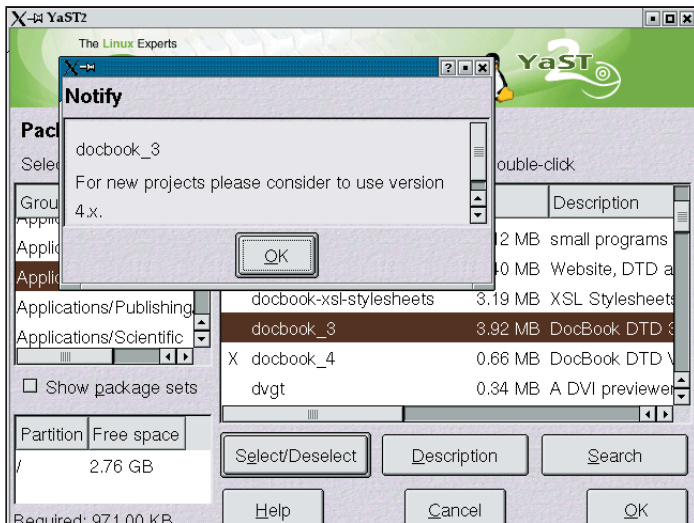
Essential information

Kernel Version: 2.4.12
X Window System: *XFree86* 4.0.1 and *XFree86* 3.3.6
Desktops: *KDE* 2.2.1, *GNOME* 1.4
Window Managers: *Sawfish*, *Enlightenment*, *WindowMaker*, *fwm95*, *mwm*, *twm*, etc.
GCC: 2.95.3
Glibc: 2.2.4
Supported Platforms: G3, G4 Macintosh or IBM RS/6000

Features: Graphical Installation, YAST and YAST2 configuration and administration utilities, *SaX2* X Window System configuration utility, etc.
Bundled Software: *KOffice* 1.1, *GIMP* 1.2.2, *Mozilla* 0.9.4, *MySQL* 3.23.41, *GCC* 2.95.3, *Apache* 1.3.20, *OpenOffice* 0.638, *Python* 2.1.1, and just about everything else
Languages: Most



YAST2 makes it easy to administer your system, regardless of which flavour of Linux you're familiar with.



YAST2 makes it easy to install new software packages, and also provides guidance during the process.

filesystems they offer during installation. They've invested in the *ResiserFS* and were the first Linux distribution to offer it as an option for root filesystems, but SuSE 7.3 also offers the *JFS* and *ext3* journaling filesystems, as well as *ext2*, the classic Linux filesystem.

Administration

SuSE has its distant historical roots in the Slackware Linux distribution, which poses some challenges to Linux fans who are used to administering and customising systems that follow the Red Hat model. Regardless of your general religious stance on command line versus graphical tools, this is the type of problem that graphical administrative utilities are designed to solve, and SuSE's YAST2 (and YAST, for the truly graphically challenged) are simply excellent.

YAST2 is a modular system that provides administrative interfaces for installing system updates; configuring and managing software, hardware, basic and advanced networking, security and general user administration; and much more. I can't say enough good things about YAST2 – it truly make it easy for someone coming from any other Linux (or Unix) system to configure and administer your SuSE system without having to first become one with all of SuSE's configuration files and administrative philosophy. The only oddity I noticed with YAST2 is that it seems to update all of its configuration information after any administrative change, not just the configuration information that you'd expect. For example, after modifying my X11 configuration information, YAST2 reported that it was updating the configuration files for things like

ypclient, news, general permissions and mount information, and *tetex*. This seems unnecessary, since I can't see what bearing the resolution of my X display has with things like NIS and *NetNews*, but I suppose that it's always better to err on the side of caution.

Upgrading/installing

Long before the Red Hat Network, YAST2 provided an integrated mechanism for painlessly downloading and installing patches and updates over the Internet. In its most trusting mode, you can simply specify automatic updating and installation and go out for an espresso or pint or two while YAST2 identifies the necessary files, retrieves them, updates your system, and prepares your dinner. As you'd expect, you can also use YAST2's update module in anal (Expert) mode, where you can pick and choose between available updates, the FTP sites from which you want to retrieve them, whether you want to save the updates on your system, and so on. Either way, YAST2 makes it easy to keep your system up-to-date with the latest security fixes and general enhancements, and essentially eliminates any excuse for administrators who let their SuSE systems go unpatched after problems have been identified by SuSE or its devoted user community.

SuSE Linux stands head and shoulders above any other PPC Linux distribution in terms of the richness of the software delivered as part of the package. SuSE 7.3 PowerPC edition comes with eight CDs packed with the binaries and sources for almost every Linux software package that I've ever used, and many that I didn't even know existed. Installing SuSE 7.3, choosing to install "everything" and watching package names and descriptions scroll by is an eye-opener in terms of the bulk and scope of software that is available for Linux today. Though I hate to admit it, I ended up taking notes during the install, not just for the purposes of this review but also in order to identify software packages that sounded like they might provide solutions to my software wish list, both personally and professionally. I was actually surprised when one of my favourite package, *komba2*, a browser for SMB networks, wasn't included – something to look forward to in the next version of SuSE.

If you decide not to go the "install everything" route, YAST2's Install/Remove Software module makes it easy to locate, select, and install additional software packages. It includes a convenient Search form

that makes it easy to identify the package in which the specific binary you're looking for is located, and even provides helpful tips if you choose to install a package for which higher revisions are also available.

Another way in which SuSE shows their devotion to their user community is in the quality and scope of their product documentation and online help. Cutting documentation has been a cost-saving strategy for many Linux distributions, but SuSE is a clear and proud exception to this sad rule. Their printed Handbook and Network guides well-written and accurate. They also provide a remarkable amount of online help tailored for their distribution, as well as the standard documentation provided with *KDE et al*.

Conclusion

SuSE Linux is still my distribution of choice for PowerPC hardware, even though there are speed bumps on the highway to Linux happiness when using it. Though I had good response from SuSE's email technical support staff, they couldn't resolve the problems I encountered when trying to install 7.3 on my older G4 tower (which makes the *Yikes!* motherboard moniker oddly appropriate). Installing SuSE 7.1 on this system worked fine, and I've reinstalled it there for that very reason.

SuSE seems to have rushed this version of its PowerPC Linux out the door a bit prematurely. If you aren't using an Airport card and don't have an older G4 with the short-lived *Yikes!* motherboard, I'd wholeheartedly recommend it as the Linux to use on your Mac hardware. If you have an older G4, I'd buy a copy of SuSE's 7.1 PowerPC edition or wait for the next revision. SuSE's obvious commitment to Linux and to the PowerPC platform makes me confident that they'll resolve the problems I encountered with 7.3. **LXF**

Linux Format VERDICT

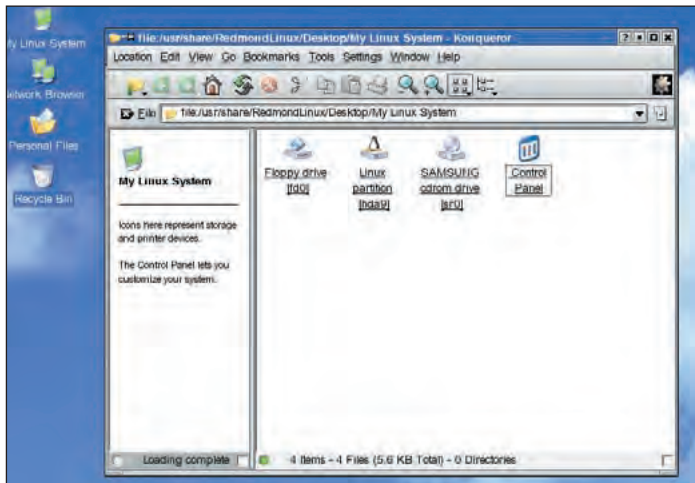
Ease of use	8/10
Features	10/10
Performance	8/10
Value for money	9/10

Still the best and most complete PPC Linux distribution with great administrative utilities, though you should avoid this distribution on specific types of Mac hardware.

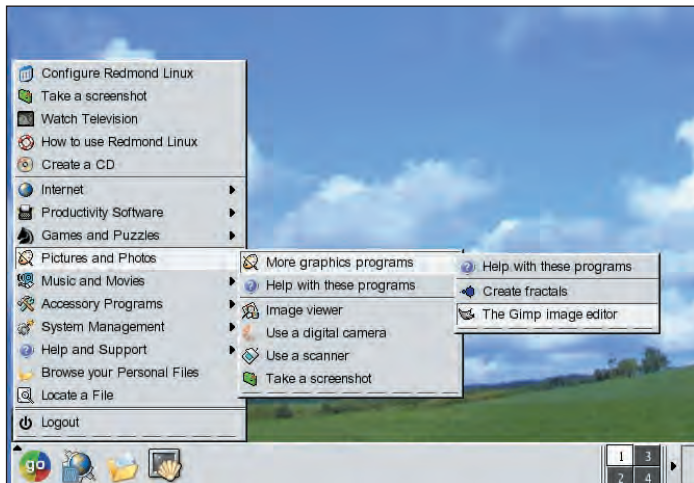
Linux Format RATING

9/10

LinuxFormatReviewsRedmondLinux



Windows users will feel at home on the Redmond Linux desktop.



Uncluttered, task-orientated menus simplify finding your way around.

DESKTOP DISTRO

Redmond Linux Personal

■ **PUBLISHER** Redmond Linux/Lycoris ■ **WEB** <http://www.lycoris.com/> ■ **PRICE** \$29.95 (1 CD version) \$39.95 (3 CD version)

Richard Drummond finds that a newcomer on the distro scene makes a surprisingly polished start at avoiding the choice newbies find so confusing.

Many Linux vendors have adopted a strategy of infiltration in the battle for the desktop. By offering a product with a user interface that looks and feels like Windows – but is built on more robust and secure foundations – they attempt to convert users to the advantages of Linux by stealth. Redmond Linux might be a newcomer, but is already playing this game better than most.

Redmond Linux is basically a clever repackaging of Caldera OpenLinux – with simplified installation, a polished and uncluttered build of the KDE2 desktop and a careful but restricted choice of application software. Everything about Redmond Linux is aimed at creating a comfortable and easy-to-navigate desktop environment for the beginner and – from its XP-like desktop theme to its built-in support for Windows networking – at making the transition from Windows platforms a painless one.

Keep it simple

Redmond Linux is available either as a single CD or 3 CD set, accompanied by a thin, A4-sized installation manual. The first disc is the bootable install disc and contains the core of Redmond Linux, while the other two discs are supplementary and contain development tools and source code.

One of the goals of Redmond Linux is simple installation, which is perhaps unsurprising for a distro aimed at new users. Thankfully – in the main – it fulfils this goal. The installer is a modified version of Caldera's *Lizard*, so, as you might expect, it is well-tested and reliable. It performs automatic hardware detection – usually leaving the user merely to confirm its configuration choices – and goes on to ask you which partition to install to, to set up user accounts, choose your time zone and so on. There's nothing out of the ordinary there. One big advantage of the Redmond installer, however, is that there is no package selection stage. Redmond Linux automatically installs all the software you need to make a useable desktop environment. This is a big plus for beginners, removing the need to actually understand all those cryptic-sounding software names.

The installer isn't perfect, however. For one, it provides no tools for the non-destructive resizing of partitions. This means that if you want a machine to dual-boot with Windows and you don't already have some unpartitioned disk space, then you will have to use a tool such as the command line *FIPS* or the proprietary *PartitionMagic* to create the necessary space. Most distros offer some kind of built-in partition resizing these days and, since

Further information

The latest release of Redmond Linux (build 44) contains the following versions of key software components:

Kernel 2.4.12
Glibc 2.2.2
XFree86 4.10/3.3.6
KDE 2.2.2
Mozilla 0.9.7
KWord 1.1

this is one area of installing Linux that is most off-putting to beginners, this definitely is a point against Redmond Linux. On the plus side it uses the journalled *ReiserFS* by default and installs *GRUB* (rather than *LILO*) as its boot loader.

Another area that needs work is the configuration of graphics cards: Redmond Linux performs this reliably if rather conservatively. For example, on one machine that I tested, it chose to install XFree86 3.3.6 to drive the Matrox G200 card that was installed in the system. Now, the G200 works perfectly well under XFree86 4.1.0, so, by opting for version 3.3.6, Redmond Linux is forcing the user to miss out on support for hardware-accelerated 3D rendering and anti-aliased fonts. A quick browse through the installer's hardware database reveals that many

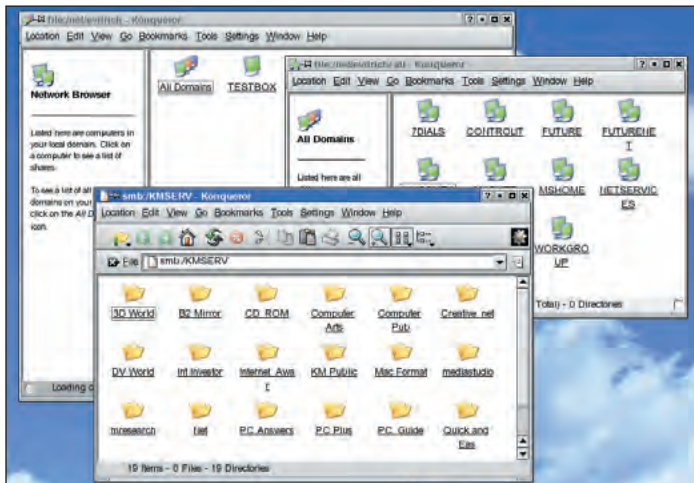
graphics cards will be similarly 'under-configured'. In addition, Redmond Linux doesn't ship with NVidia's XFree drivers – so there will be no built-in 3D acceleration for owners of TNT and GeForce cards either. These problems are not insurmountable and, in fact, are not that difficult to fix, but are not something that a new user should be expected to deal with.

Desktop tidy

Once you boot up into your newly installed Redmond Linux and log in, the first thing you notice is how like Windows it is. For instance, they have adopted the *MyComputer* and *Documents* metaphors and even the trashcan is called the 'RecycleBin'. Clicking on the 'My Linux System' icon on the desktop launches a browser window with links to all your drives and to the *Control Panel* (or *KDE Control Center*), while the 'Personal Files' icon links to a folder in your home directory for storing, well, your personal files. The latter would be a nice touch if applications actually defaulted to storing files there. After all, the root of your home directory is wont to get rather cluttered with all the junk that tends to get stored there, so having a default folder for saving documents to is a good idea.

The second thing you notice about Redmond Linux is how well laid out

LinuxFormatReviewsRedmondLinux



Samba is well-integrated with Redmond Linux, allowing automatic participation in a Windows network.

things are. The familiar KDE menu, for instance, has been re-ordered and re-labelled to make navigation easier for the beginner, and a better use of hierarchy means the user won't be overburdened with choices all at once. Menu entries are predominantly task orientated — which again is clearer for beginners. It is much more obvious that the option 'Use a scanner' loads up the scanner interface rather than if it were labelled with the cryptic program name SANE.

The KDE Control Center has also undergone a similar reorganization, with the individual control panels more clearly ordered in the tree and with the less frequently used panels tucked away in nested branches. Various other configuration tools have been integrated into the Control Center, too. These include graphical tools for configuring your X driver, your network configuration and so on, and bringing everything under the umbrella

of the Control Center makes finding the tools you need much easier.

Two key points about the Redmond Linux desktop for Windows users are the integration of Samba and WINE. Samba is installed and configured automatically, and lets you communicate with a Windows network right from the word go. The graphical network browser fulfils the function of the Windows Network Neighborhood and lets you browse and access remote Windows shares, whilst your own home directory is automatically exported for sharing by Samba. One complaint is that there are no tools to configure how your home directory is shared. The built-in version of WINE is also useful for those who need Windows compatibility and lets you launch Windows applications from the desktop. Again there are no configuration tools for WINE, though.

In terms of style and presentation, Redmond Linux impresses. Its look

and feel is so much more professional and polished and a lot less *ad hoc* than other distros. What is disappointing is the lack of application software. In one way, this a good thing. Users are not bewildered by having three different browsers, four paint packages and half a dozen MP3 players — and the choices that Redmond Linux have made are, by and large, sound. Mozilla is the default browser — a sensible option considering that it works better with the Flash, RealPlayer and Java plug-ins provided, and gives an all-round better browsing experience. Redmond Linux also offers KMail for email; GIMP for graphics work; XMMS for playing multimedia files. The only real weak point is KOffice. While KWord is much more stable and useful than it used to be, it is still not in the same class as MS Word, and users converting from Windows will find it wanting.

Applications

Making up for the shortage of applications by installing new software on your Redmond Linux machine is not as easy as it should be. While there is an update wizard to automatically download and install applications from a Redmond Linux mirror site, that is all it currently does — update software; it does not offer to install new software. The desktop menu entry promisingly entitled 'Install software' simply launches a browser window with a form for you to fill in to request new software in Redmond Linux. The second and third CDs from the three-CD set aren't much help either. They auto-run when inserted in your drive, but just launch a browser window. Okay, you can then use KPackage to install them, but this doesn't resolve dependencies — and

besides these contain only source code and development tools. There is a fourth CD image, which you can download from Redmond's website, and this does contain more actual application software. Why can't they provide an install wizard to access these extra packages?

The fact is, Redmond Linux accomplishes what it sets out to do. It provides a stable and useable desktop environment that will be easy for Windows users and those new to Linux to get to grips with. Despite the paucity of software, what is provided will cater for the needs of the majority of users. Redmond Linux is a long way from being perfect, though. The online help — via the KDE Help Center — does not provide useful documentation for all the software installed and the lack of any partition-resizing tools means that it is less appealing for new users than it could easily be — and the problem with configuring graphics cards is annoying. These issues can be easily fixed.

Redmond Linux is undergoing a name change, so the next version will be under the new name, Lycoris. **LXF**

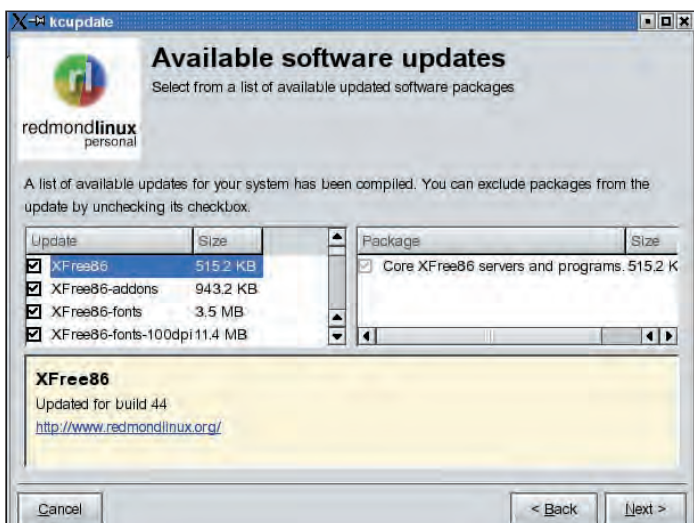
Linux Format VERDICT

Ease of use	9/10
Features	7/10
Performance	7/10
Value for money	8/10

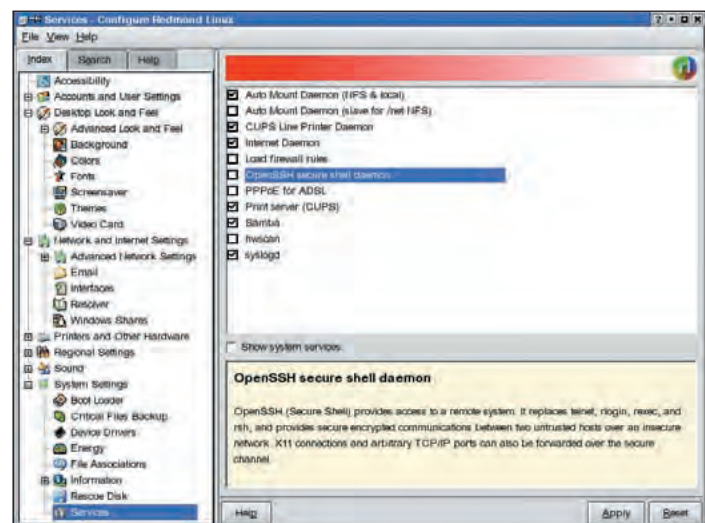
Needs tweaking in some areas & more apps, but Redmond Linux is a polished, well-thought out desktop distro.

Linux Format RATING

7/10



The update wizard checks online for updates to installed software.



Redmond Linux's config tools are intergrated into the KDE Control Center.

MAIL SERVER

Caldera Volution Messaging Server

■ **PRICE** \$899 (for 25 mailboxes) ■ **PUBLISHER** Caldera ■ **WEB** www.caldera.com

Keeping things simple and low-maintenance in your office? **David Coulson** examines an all-in-one mail server solution with a simple and straightforward web-based interface.

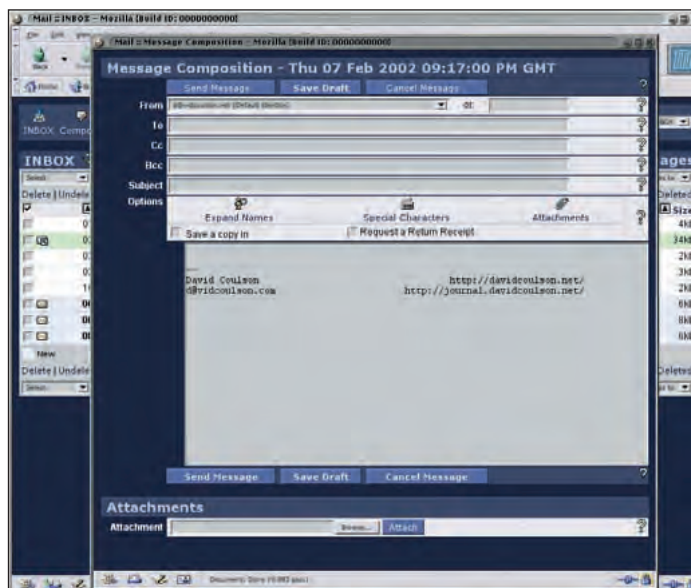
Email within business is now a necessity and is often used as a primary method of communication between a business and their customers.

However, for small to medium sized businesses, which have outgrown ISP provided services, yet do not have a dedicated IS department to manage a mail service, finding the right product can often be a long, drawn out, and costly process. Caldera, the company well known for their OpenLinux distribution, have come up with an 'all in one' package for anyone needing an easily managed and stable mail server.

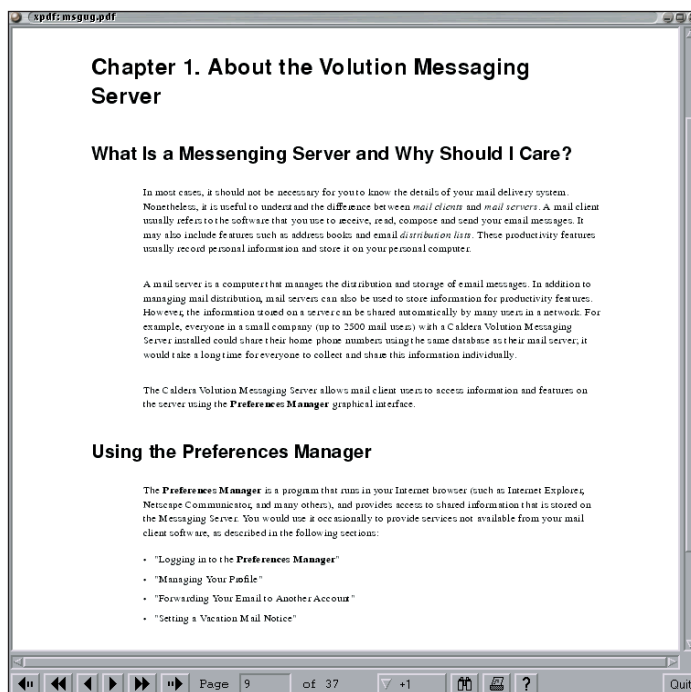
The Caldera *Volution Messaging Server* offers a wide variety of features and capabilities, which enable even the most IT-unfriendly company to have a mail server up and running quickly. It does, however, require that you have some variety of dedicated Internet connection to host the mail server on, but that is outside of the scope of this review. Caldera *Volution* is purchased with a specific mailbox capacity, so the more people you have using the system, the more it is going to cost you.

The installation of *Volution* is RPM based, but it is only realistic to install it under either OpenLinux or Caldera's Open UNIX platform (which has a Linux Kernel Portability system, so you can run regular Linux binaries on it). It won't work with Debian or even RedHat, since, even if your distribution uses RPMs, *Volution* relies on many OpenLinux-specific features.

We tested *Volution* with Caldera OpenLinux 3.1. This comes with a default kernel requiring Pentium II-based hardware, so if you've an old 486 or P100 which you wanted to make use of, you're going to be disappointed. Naturally, this limitation reduces the viability of *Volution* for many businesses who want a very low-cost option, but if you're going to be buying new hardware for it, then it will not be a problem.



Web based email courtesy of IMP – collect your mail on the move.



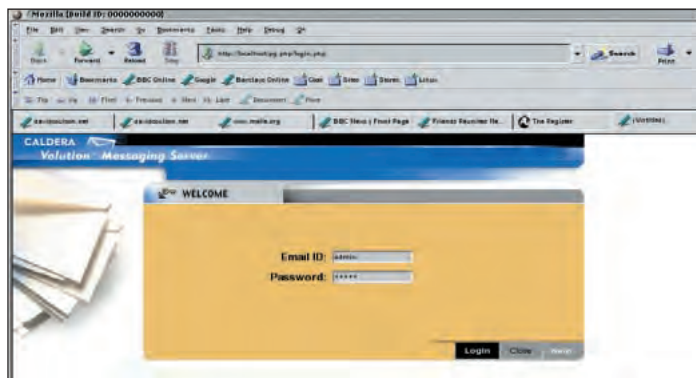
Volution Messaging Server has particularly good docs for the confused.

Installing *Volution* on OpenLinux is straightforward, as it auto-detects the insertion of a CD, and runs the installer automatically. While *Volution* itself does not need X to run, the installer does. As you would expect, the installation must be performed as root, so that the RPMs can be installed successfully. The installation does little other than check that you comply with the license agreement, and takes next to no time to complete.

Once it is up and running, you access the *Volution* system via a web browser, and all management procedures can be performed remotely. You can start by adding domains, so the server knows what it should accept mail for, and what not to. Within a domain you can have either users or aliases. *Volution* also supports virtual domains for mail, so you have multiple domains handled by a single network connection. Using virtual domains is slightly different, as you need to structure your users properly – so they apply to a specific mail domain, rather than just using their regular user name – but everything else is fundamentally the same.

User details

User information is stored within a LDAP database, which is accessed by *Volution*, as well as the IMAP and POP3 servers it uses, so when you create a user using *Volution*, you are not really creating a user on the server, only including a reference for them within the LDAP directory service. *Volution* also supports quotas for user mailboxes, so you can restrict the amount of disk space your users can gobble up with their email, as well as specifying if attachments are removed from email, further reducing space. Quota support can be very useful if managed properly, otherwise people will wonder why they don't have any new emails, and it's not a wise option to have it reject email with a quota



The **Volution** server web interface – configure your system with ease.

message, as I'm sure customers would love to hear that.

As well as the admin, specific users can be granted privileges to change their profile information, and to create new aliases within domains. While this means they cannot completely manage the mail server, it removes many of the simple and tedious exercises from the administrator. Regular users can also view the profiles of other users, as well as change their password via the web interface. A rather useful addition is that users can find out exactly what they need to configure their mail client with, so they can access their mail and the directory services, which saves a great deal of time educating users on mail client configuration and greatly improves the ease with which new users are added to the system. Most businesses will have many basic email addresses, which have the mail delivered to all of their staff in that specific department, and *Volution* handles mail aliases as simple mailing lists. This gives a very flexible structure to mail delivery, and you can easily create lists where only the subscribers can post to it, which is very useful for an internal office list, or if you want to have a managed announcement list for your customers. It is not possible to remotely subscribe to the mailing lists, as you need to edit the alias list directly, so it is not suitable if you want people to sign up on their own.

While *Volution* handles mail services, it also functions as an LDAP directory service, so mail clients, such as *Outlook*, *Mozilla* and *Netscape*, can use it to look up email addresses when composing email or for general searches for information. If you've ever needed to look for a colleague's phone number, or find out who is in charge of a specific department, then you'll appreciate the use of a

centralised resource for storing everyone's details. You can, of course, use a variety of PIM applications to access this information too, so it can generally be completely integrated into any office organisation policy which already exists. Users of *Outlook 98* and above can have *Volution* automatically configure their mail client for them, which when you have a few thousand users, few of which understand email, is a very pleasant prospect. *Volution* can work with any IMAP4/POP3 client and, indeed,

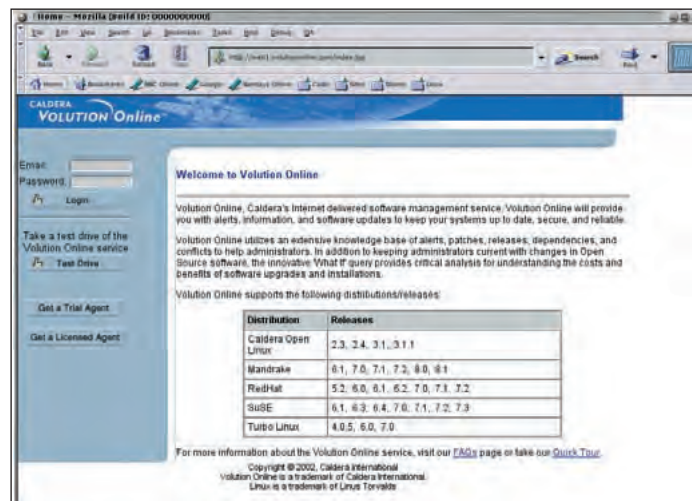


OpenLinux has *IMP* installed as default, which is a very simple to use, yet fully featured web-based IMAP mail client, so even when people are out of the office, or are not at their own machines, they can pick up their email.

Security is a major consideration once people begin to do things outside the local office network, and *Volution* has SSL/TLS capabilities, using *stunnel*, for its IMAP and POP services, and the SSL capabilities of *Apache* with **mod_ssl** for the configuration interface and web mail.

For those who don't like web interfaces, there is a command line based tool to create new users and configure mail preferences, and you can generally simply edit basic configuration files in order to change the capabilities of your mail server, which will come as a welcomed alternative for any poor soul who has tried to battle with *Sendmail's* rather unpleasant m4-based configuration system in the past.

Viruses and unsolicited commercial and bulk email (UCE/UBE) are major concerns for business and, indeed, any users, and as *Volution* uses *Postfix* for SMTP mail delivery, you can easily filter email based upon any header. As with any Internet facing service, *Postfix* has external relaying disabled as



Volution Messaging Server can be upgraded using Caldera's '**Volution's** system, which is useful for security.

default, so you can be assured that your own machine will not be used as an open relay to further distribute spam or unwanted messages. *Volution* also supports the use of a number of different anti-virus systems, which can scan incoming and outgoing messages for well known virus signatures, but the anti-virus package must be kept up to date in order to get the most benefit from it. There is also support for filtering, and mailboxes under IMAP, for individual users, so organising your messages and ensuring that they all end up in the right place. This is, of course, in addition to any filtering which is performed locally by the mail client.

Worth the money?

Volution is a very neat, and well put together package. As it is based almost entirely upon Open Source, and freely available software, the more advanced user can generally duplicate the same capabilities, sans the web-based configuration tools, with little effort and a much reduced cost. However, *Volution* and Open Linux are aimed at those who want to get things done with the shallowest learning curve and little ongoing maintenance.

\$899 for twenty-five mailboxes is by no means cheap, and for companies with a few hundred employees, it can quickly spiral out of control, at \$2,799 for a one-hundred user increase. Considering you are really paying for the web interface, and a neat package of everything else, at times when many companies are tightening their belts and trying to weather the general economic difficulties everyone is facing, it may be difficult to justify such a cost when an IT consultant could provide a similar system, using entirely open source

tools, for less – and with far more flexibility.

The limitation of only being able to install *Volution* successfully on OpenLinux systems makes migration and server consolidation somewhat more difficult, and if you already have Linux machines in your office, it may be an unwelcome something new for someone to learn about.

One must also bear in mind that as well as the *Volution* system, there is also general system maintenance, in particular DNS, to consider, which requires a little more understanding than just clicking on things. Caldera recommend using *Webmin*, under OpenLinux, to admin your DNS server, but it still does require an understanding of DNS and networking, so it is not a completely novice solution. However, the ease of use of *Volution*, considering the breadth of its features, more than makes up for spending a little time learning how to do everything else properly.

If you're after a quick and capable mail solution, then *Volution* should be at the top of your list. **LXF**

Linux Format VERDICT

Ease of use	8/10
Features	9/10
Performance	8/10
Value for money	7/10

Caldera have managed to combine a selection of Open Source services into a very easily maintained system, offering a wide range of features.

Linux Format RATING

8/10

WEB DEVELOPMENT ENVIRONMENT

Quanta Gold

■ **DEVELOPERS** theKCompany ■ **PRICE** \$49.95 (Download \$39.95) ■ **WEB** <http://www.thekcompany.com>

Does adding extra features to a free project make a commercial hit? Jono Bacon investigates theKCompany's claim of "web development made easy."

Quanta Gold is a web development environment produced by theKCompany. The already successful *KDE* application *Quanta Plus* was taken by theKCompany and improved and developed. theKCompany also converted *Quanta* to pure *Qt* (and not using the *KDE* libraries) so it could be distributed for not only Linux but for other *Qt* supported platforms including Windows and eventually MacOS.

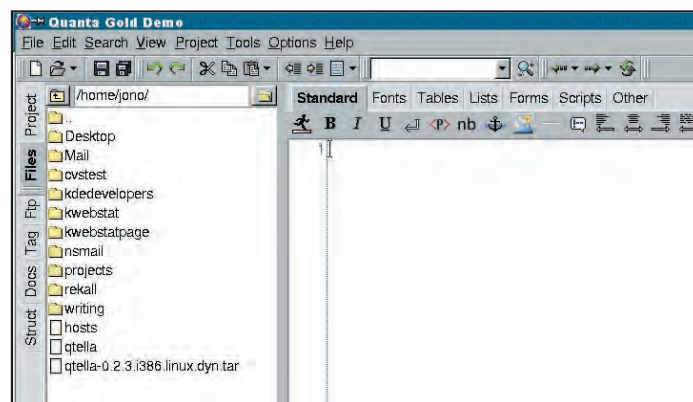
Now there is a commercially available edition of *Quanta*, how does it improve over the free *Quanta Plus* and other editors which are available? Well, interestingly enough, *Quanta Gold* offers many features:

- Syntax highlighting with support for *ColdFusion*, XML, PHP, SQL, Python, Perl, DHTML, *Zope*, C++, HTML
- Integrated FTP client.
- Point-and-click interface to give easy access to a wide range of preformatted code fragments and attributes.
- Contextual attributes dialog — press the F4 button at any time, and you're presented with a list of possible attributes for the currently selected item or function. Choose one, and it's immediately integrated.
- Tree view to allow you to elegantly check and manipulate your website's structure.
- Preview function lets you see the results of your work while you're working.
- Easy upgrade path from previous versions of *Quanta*. You can natively use *Quanta* project files in *Quanta Gold*.
- Contextual documentation included for scripting languages such as PHP, JavaScript and CSS. Just press the F1 button when the cursor is in any particular PHP function or HTML fragment to get the help you need.
- Integrated CVS support

As you can see it has quite decent features but the true test is in practice.

Installation

Packages were available in various formats including RPM and .tgz. Most popular distributions such as RedHat,



The main window — your website starts here.

Mandrake, SuSE etc have prebuilt packages available. Debian packages are also planned.

I decided to use the .tgz package to install *Quanta gold* on my Debian machine. I unpacked the archive and there was an *install* script I could run to install the application. This was a fast procedure and I was up and running straight away. The only problem I could foresee in my installation was that *Quanta Gold* was not added to my *KDE 'K'* or *GNOME* starter menus.

As a side note, I tested the Windows edition of *Quanta Gold* to make some comparisons to the Linux edition, and there was a graphical installer supplied. This certainly made installation easier and more attractive,

and it would be nice to see a Linux equivalent in a future release.

Getting started

When you load *Quanta Gold*, the main window pops up after a nice little splash screen and first impressions are certainly positive. *Quanta Gold* appears quite similar in visual design to *HomeSite* on Windows based systems (many HTML editors, such as *Bluefish*, have also attempted to emulate the good design of *HomeSite*). *Quanta Gold* not only borrows from this design paradigm, but also adds to it with tabs on the left side of the screen to offer access to your files, workspaces, FTP upload et al. This makes accessing various parts of your work easy and straightforward. Another factor in

judging *Quanta Gold's* visual qualities, and some may see it as a minor consideration, is that it looks a generally attractive application. It uses the *Qt* toolkit, which fits in well to the *KDE* environment if available, and does not use *Motif* which many feel has an old fashioned and somewhat clunky feel. Looking around the main window we see:

- Tabs: for access to Tags, Structures, Documents, FTP, Files and Projects.
- The sidebar which shows whichever tab you selected
- The tag toolbar which has facilities to add a large amount of tags easily
- The main editor

An interesting area to look at is the tag toolbar. This provides a selection of toolbar buttons which you can click to insert the relevant tags; certain items will also ask for attributes you can set when you add a tag. *Quanta Plus* offers a good selection of functions, but *Quanta Gold* has a huge amount more — in particular scripting support has been improved with a choice of toolbar options for PHP and JavaScript among others. Also there are many database functions that can be inserted into your code, with support for most of the major databases.

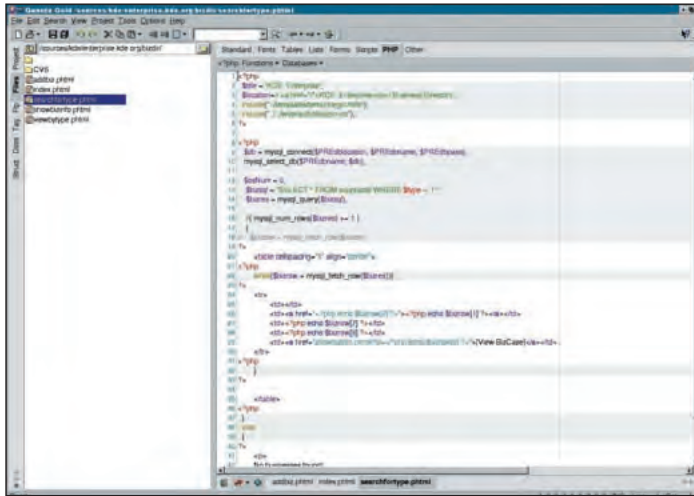
Creating some code

Using *Quanta Gold* is pretty easy, and before long I was writing code. There are a great number of syntax highlighting styles, and the highlighting is clear. Block highlighting is used for scripted code which helps differentiate between markup and scripts.

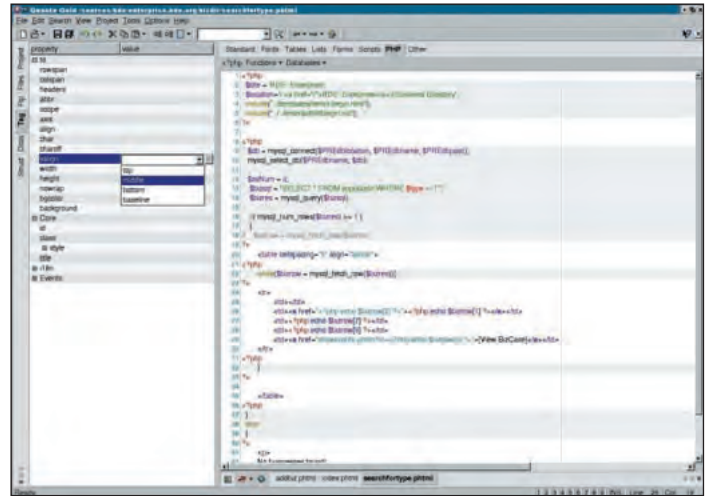
The editor is nice, and contains a margin down the left side for line numbers. Although only a small feature, the line number margins can be useful for discussing code with other developers — particularly when source management tools such as *CVS* are used. Auto indent, adjustable tab size, custom font settings and backspace indent make the editor comfortable to work in.

While editing your code, there are some interesting facilities which can be utilised. One feature is that if you put the cursor on a tag and press F4, a dialog will pop up for you to tweak the tags' attributes. This can be handy when you can't remember particular attributes. Another useful feature is word completion — type a few letters, then press Ctrl-Space and a little box will pop up with tags that match those letters.

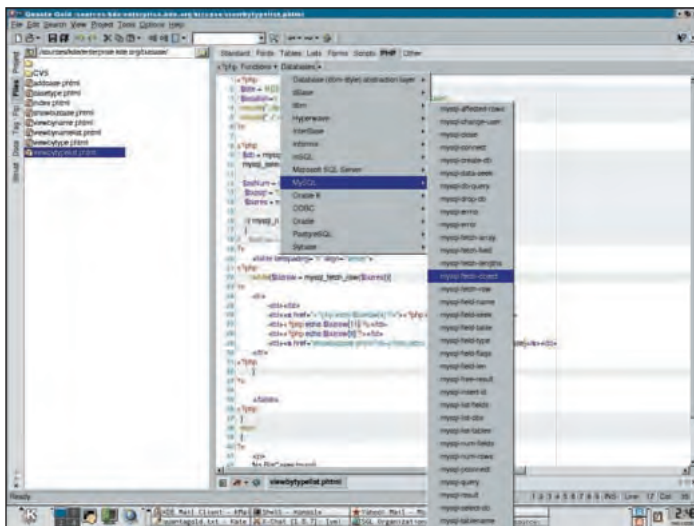
Some other facilities that will help productivity can be found in the tabs on the left hand side — one of the most useful being the tag properties editor. Each tag that your cursor is on



Quanta Gold editing some code showing the highlighting.



Setting tags using the sidebar — very configurable under Quanta Gold.



Using the database tag support on the tag toolbar.

will bring up the attributes in that tag on the left sidebar. This gives direct access to all the facilities of that tag. There is also a tab for showing the structure of the page which is similar to the Document Object Model.

Source management

An increasing requirement in web development houses is a suitable system for source code management; this is probably a situation where many developers are working on the same codebase. CVS is often used in these circumstances, and CVS integration is included within *Quanta Gold*. This is an important feature not only for the Linux edition of *Quanta Gold* but also for the Windows edition where there are few decent CVS clients. This also makes source management fairly transparent to the developer. It would be nice in future versions of *Quanta Gold* to provide some kind of web interface to the CVS server — something along the lines of cvsweb,

but also with direct ability for committing and viewing logs.

On the whole the CVS support is pretty good, and works as I expected it. Menu options for committing, adding, viewing logs and other CVS duties are available, and output of CVS is shown in the message window. There were a few occasions where I would have liked to right click a file and be able to commit it contextually or update a directory; I hope these are usability issues that will be eliminated in future releases.

FTP support

Quanta Gold has FTP support built in for uploading your web pages to your webserver. When you click on the FTP tag, the sidebar on the left is filled with a panel in which to set up your FTP server connection and pre-existing connection accounts.

The ability to have a number of FTP servers makes the uploading of multiple projects to different

webservers fast and easy. There are remote and local panes that allow the transfer of files cleanly and easily. A nice feature is that files can also be viewed from the remote side. The only snag I noticed was some problems with viewing images remotely within the application.

Comparisons

Quanta Gold in comparison to *BlueFish* is fairly simple. *BlueFish* offers some interesting facilities such as WML support, but also lacks things such as CVS support. *BlueFish* is less user friendly in operation and *Quanta Gold* makes integration of large projects with many files easier. *BlueFish* however does have custom menu support which would make custom commands easier to produce, and this is something theKCompany should make note of.

In comparing *Quanta Gold* to *Quanta Plus* we essentially have a very similar interface (apart from the left tabs in *Quanta Gold*). *Quanta Plus* is a very capable editor, although it does lack decent support for scripting languages such as PHP and also languages such as JavaScript. *Quanta Plus* does behave in a very user friendly way and certainly makes web development a comfortable process, but unfortunately the lack of features that *Quanta Gold* makes up for means *Quanta Plus* fall short.

Conclusion

Quanta Gold is a powerful and well designed application. theKCompany have done a good job of adding and refining *Quanta Plus* for professional use. I am sure *Quanta Plus* will continue development, but the option of *Quanta Plus* and *Quanta Gold* should make a nice choice for the user.

There is no doubt that *Quanta Gold* offers a number of features that are worth paying for, and considering you get *Quanta Gold* for Linux, Windows and other systems, it is good value for money. *Quanta Gold* would find particularly good use in web development houses where multiple operating systems are used and a common application is required for training. The integration of CVS is nice and the ever increasing use of CVS in web development houses really justifies this feature. The support for scripting languages is wide, although ASP would be nice. Another additional feature that would be nice to see in *Quanta Plus* is a WYSIWYG (What You See Is What You Get) mode for creating pages graphically akin to Macromedia *Dreamweaver*. This addition, and maybe some wizards for creating tables and forms, would really make *Quanta Gold* an all rounder.

In general usage I found *Quanta Gold* an extensible and useful application, and in conjunction with PHP, Apache, CVS, MySQL *et al.* it really tops the developer toolset off to make Linux a capable and prominent web development system. **LXF**

Linux Format VERDICT

Ease of use	8/10
Features	8/10
Performance	8/10
Value for money	9/10

Quanta Gold is a powerful and flexible web development environment. Decent features combined with an easy to use interface make this one recommended.

Linux Format RATING

8/10

XHTML Essentials

■ **PUBLISHER** Wiley ■ **AUTHOR** Michael Sauers,
R. Allen Wyke ■ **ISBN** 0471417645 ■ **PRICE** £32.50

Tireless reader Richard Drummond tackles an introductory tome on moving to XHTML.

HTML is dead, long live XHTML – at least, that is, according to the W3C, the body that maintains the various web standards such as document mark-up. *XHTML Essentials* aims to help the web developer make the transition from lazy and forgiving HTML to the stricter, XML-conforming XHTML.

The book begins by exploring the history of HTML and explaining the *raison d'être* for XHTML. It then launches it an overview of basic XHTML tags, followed by chapters on creating XHTML documents, validating XHTML and translating HTML to XHTML. Subsequent chapters cover

XHTML tables, forms and frames in details. Code snippets and screenshots help illustrate how the various tags are used.

Later sections are more ambitious, covering advanced topics such as Cascading Style Sheets and creating dynamic pages with Javascript and the Document Object Model. These chapters provide a good introduction to these subjects, but necessarily the treatment is not as in-depth as would be provided in dedicated volumes. Useful code extracts and recipes are provided to solve common problems, such as roll-over buttons, and the material covering the differences

between common browsers and how to write your code accordingly is particularly helpful.

The main problem with *XHTML Essentials* is that it fails to live up to some of its claims. For example, the back cover states it is 'the complete how-to guide to moving Web development from the desktop to wireless technologies'. I think not. Nevertheless it is not a bad book. It forms a solid reference to XHTML that provides a useful and more readable companion to the W3C's

specification. It also does well at illustrating just where XHTML fits into the larger world of web development.



Linux Format VERDICT

Makes a good reference for developing in XHTML, but lacks depth and focus.

Linux Format RATING

7/10

Designing from Both Sides of the Screen

■ **PUBLISHER** New Riders ■ **AUTHORS** Ellen Isaacs,
Alan Walendowski ■ **ISBN** 0672321513 ■ **PRICE** £34.99

Richard Drummond is impressed by a practical look at designing a better user interface.

Many software applications have poor user interfaces, simply because more effort is spent on ticking off items on a features list rather than thinking about the tasks a user needs an application to perform. This book aims to redress the balance – to create what the authors call cooperative technology. The 'Both of the Sides of the Screen' of the title refers to the fact that one of the authors is a user interface designer, the other a software engineer.

The first section of the book spells out the dos and don'ts of good user interface design – with plenty of real world examples of software, web sites

and embedded applications which fall foul of these principles. These rules seem common sense in hindsight – such as making frequently used tasks easily accessible to the user and not overwhelming the user with information – but it's surprising how often they are ignored.

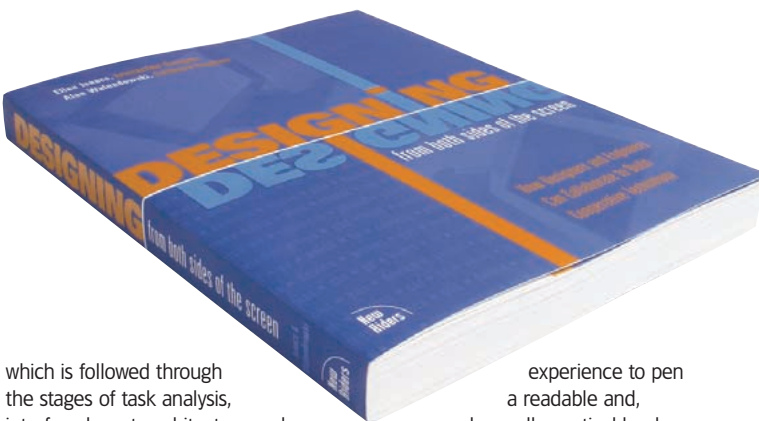
The book's second section documents the process of engineering cooperative software and shows how user interface design should be integrated with the traditional software life-cycle. This is illustrated by an account of a concrete software project – the design and implementation of an instant messaging system called Hubbub on Windows and the Palm –

which is followed through the stages of task analysis, interface layout, architecture and implementation.

It is highly illuminating to see how the design principles advocated in the first section can be applied in practice and especially to see the progress and mistakes made. I found the material on dealing with the conventions and restrictions of different platforms when doing cross-platform development – and the section on the network architecture of Hubbub – particularly instructive.

In *Designing from Both Sides of the Screen* the authors have successfully drawn on their collective

experience to pen a readable and, above all, practical book – which makes a refreshing change from the usual dry tomes on the theory of human-computer interaction.



Linux Format VERDICT

An immensely practical and enlightening book that shows how to make the user the focus of software engineering.

Linux Format RATING

9/10

Programming Linux Games

■ **PUBLISHER** No Starch Press ■ **AUTHORS** John R. Hall and Loki Software
 ■ **ISBN** 1886411492 ■ **PRICE** £25.99

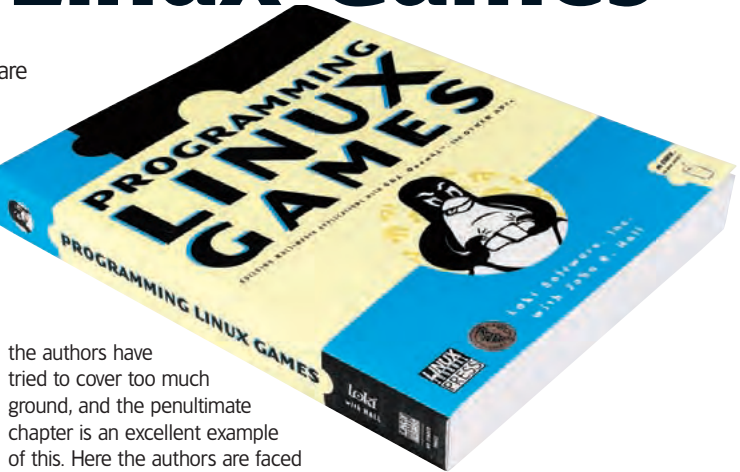
Richard Drummond discovers Loki's last outing is a book which will get Linux games written.

Open Source software has put games programming back within the reach of the bedroom programmer – but just how do you get started? This is the question addressed by this book, authored by Linux games maestros, Loki Software. *Programming Linux Games* primarily covers the tools and APIs available to the wannabe games programmer: it deals with the systems aspect of writing games on Linux. If you need information on crafting a realistic 3D engine or on AI routines, you best look elsewhere.

The book assumes some knowledge of C programming, but begins gently with the architecture of a typical game and a roundup of

common tools and how to use them before giving a broad overview of the various APIs on Linux that are suited to games programming. The subsequent chapters cover some of the more popular APIs in more detail – including SDL, OSS, ALSA, and OpenAL – while other chapters deal with topics such as creating networked games, game scripting (using Tcl) and using packaging tools such as *RPM*. These subjects are usefully illustrated throughout by an ongoing project – a parallax-scrolling, shoot-em up called *Penguin Warrior*.

Programming Linux Games is written in a light-hearted style which belies the quantity of information that it conveys. However, I think perhaps



the authors have tried to cover too much ground, and the penultimate chapter is an excellent example of this. Here the authors are faced with the task of turning the unfinished *Penguin Warrior* into a functioning game – and in their hasty rounding-up, they gloss through the potentially difficult-to-grasp subject of collision detection. Nevertheless, I highly recommend this book to the novice games programmer. In particular the sections on SDL and the under-documented kernel frame-buffer API are essential reading. **LXF**

Linux Format VERDICT

A digestible introduction to the APIs available to the Linux games programmer, but you'll soon outgrow it.

Linux Format RATING

8/10

BROWSER PLUGIN INTERPRETER

CodeWeavers CrossOver 1.0.1

■ **DEVELOPER** CodeWeavers ■ **WEB** www.CodeWeavers.com ■ **PRICE** US\$19.95 download version

Hoyt Duff gets the best of both worlds, with proprietary formats on Linux.

Ever hear of the Sorenson Video 3 codec? Neither had I until I wanted to view a movie trailer on my Linux computer – and found it it couldn't be done. That is,

until I installed an interesting app from CodeWeavers and significantly improved the functionality of my *Konqueror* browser.

Known for their work on *WINE* and

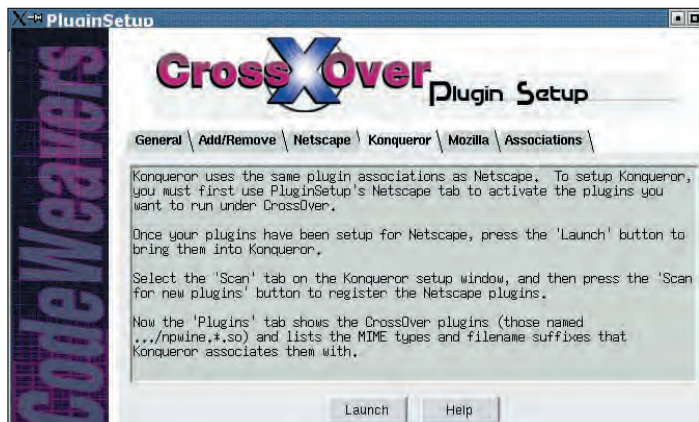
their *WINE* config tool, CodeWeavers have extended their use of *WINE* in a unique application. *CrossOver* allows the use of MS Windows plugins with the *Netscape*, *Mozilla*, *Konqueror*, *Galeon* and *Opera* browsers.

Not yet impressed? Not only is it currently the only way to get a Sorenson-encoded *Quick Time* video to play in Linux (sans a more expensive *VMware/Win4Lin* solution), it also allows you to install plugins to view *Word*, *Excel* and *PowerPoint* files as well as Shockwave sites. You can

view those troublesome email attachments in your browser as well.

The catch is that you have to pay for *CrossOver*, available from their site via credit card payment (US\$29.95 for the CD version).

The web site offer a list of supported plugins, online documentation and a very informative FAQ gathered from beta testers. Read it all before buying to discover any issues you may encounter. **LXF**



Konqueror works with *Crossover* as well, just not as automatically. Use the Launch button shown here to access the *Konqueror* plugin config screen.

Minimum Requirements

200+ MHz CPU (400+ MHZ for *QuickTime*)
 glibc 2.1 or greater
 X11R6 3.3 or greater
 Supported browser

Linux Format VERDICT

Ease of use	9/10
Documentation	10/10
Performance	9/10
Value for money	10/10

An excellent application of *WINE* technology; software worth paying for.

Linux Format RATING

9/10

Roundup

Every month we compare tons of software, so you don't have to!



Our selection at a glance

- webCDwriter
- CDBakeOven
- GCombus
- GnomeToaster
- KOnCD
- KreateCD
- XCDRoast

As you are probably aware, you cannot write to CD-ROM as transparently as you can to other media. Some special software is required to control the CD burner, and, in the case of data tracks, an image of the filesystem must be prepared beforehand. Linux has a good selection of open source tools to these jobs, including *cdrecord* and *mkisofs* (both part of the *cdrtools* package) and *cdrdao*.

All of these tools are shell-based, however, often a requiring a complex invocation from the command line. They are perfect for the batch-process building and writing of CDROMs, but for those one off quick tasks they can be just too cumbersome. Inevitably, this fact has lead to the creation of various graphical front-ends for the tools *cdrecord*, *mkisofs* and so on.

Which one should you choose, though? Which is the most suited to your own needs? Find out as we put seven of the most up-to-date CD writing applications through their paces and compare their features and ease-of-use.

Writing CDs under Linux needn't be difficult. **Richard Drummond** compares some applications which take the heat out of CD burning.

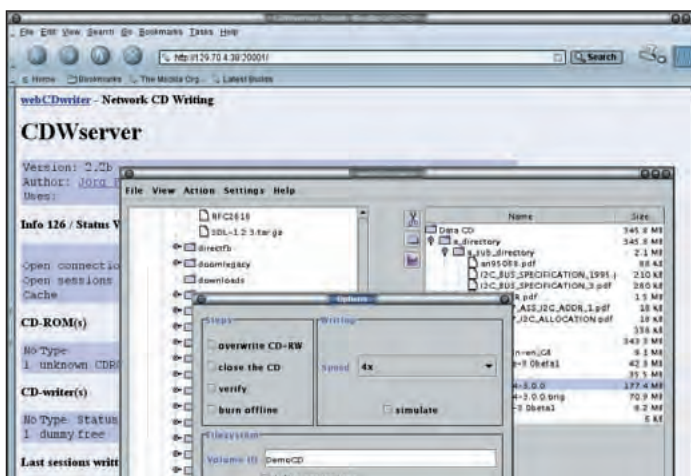
Most desktop PCs these days are equipped with CD writers, so it is only natural to expect to be able to use

them under Linux. Thankfully, the Linux kernel has driver support for the most common types of device, including SCSI, ATAPI, USB and

parallel port varieties. Configuring such devices for use under Linux is not difficult and most modern distros do this for you.

webCDwriter

■ Version 2.2b ■ **WEB** <http://www.homes.uni-bielefeld.de/jhaeger/webCDwriter/>



If you need to share a CD writer, *webCDwriter* is the tool for the job.

webCDwriter is different from the other apps on test. It is a distributed solution – consisting of a server that runs on the machine with the CD burner and a GUI-based client which lets you master and burn CDs remotely. This client is a Java Applet: it will run on any Java-enabled web browser. *webCDwriter* is great for an office environment and lets you share one CD burner between many staff.

Installing the server is straightforward and it can be configured with a simple config file. The client can be more difficult to set up depending on what browser you have. One problem is that the client needs more permissions than Java Applets usually get, so the applet is signed – but alas by a non-standard CA – so you need to install the CA's certificate as well.

webCDwriter supports modes for burning from an ISO file, mastering data CDs and mastering audio CDs. Its main interface provides a double list view where you can, depending on mode, select source files for the CD. In

audio mode this lets you pick WAV and MP3 files as tracks for the CD; in data mode you build files and directories into a filesystem tree. It's simple, elegant and works well.

Despite lacking some tools that the others have – it cannot be used to copy or rip tracks – *webCDwriter* is remarkably functional. It supports El Torito, multi-session discs and has tools for blanking CD-RWs.

Linux Format VERDICT

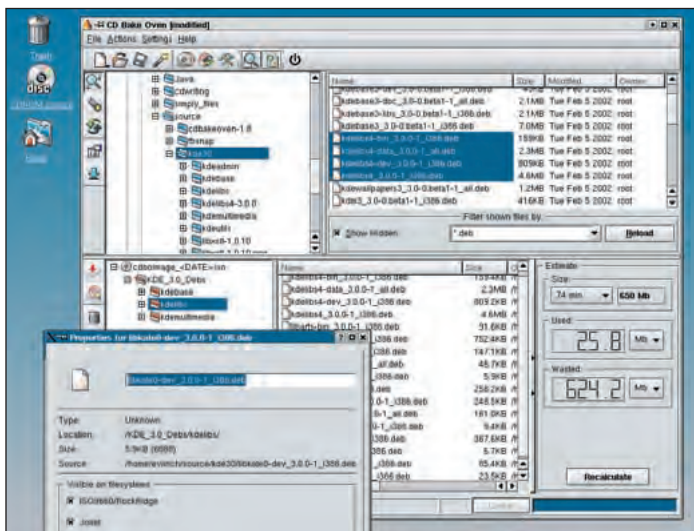
Installation	6/10
Documentation	8/10
Features	7/10
Ease of use	9/10

Not as feature-packed as some of the non-distributed CD writing software, but well executed and very useful.

Linux Format RATING
 8/10

CDBakeOven

■ Version 1.8 ■ Web <http://cdbakeoven.sourceforge.net/>



CDBakeOven's tools for building ISO filesystems are second-to-none.

Despite the high-appearing version number, *CDBakeOven* in its current form is unfinished – it is in fact derived

from an earlier Java-based project. Much is still yet unimplemented, but *CDBakeOven* boasts some unmatched

tools for mastering data tracks. (Audio support, on the other hand, is either not complete or very buggy, and we had little success with it during testing.) Unsurprisingly, then, its user interface can look rather busy and the absence of documentation doesn't help, but if you persevere it can be quick to use.

When you start up *CDBakeOven*, it presents you with a simple dialog offering options to either copy a CD, burn an ISO image to disc, create an audio CD, erase a CD-RW or open the settings editor. The first two of these are indeed as simple as they look and provide quick access to these frequently used functions.

CDBakeOven's main interface is necessarily more complicated. It goes for the three-panel approach with your file system tree at the top left, the contents of the current directory on the top right, and the contents of the ISO filesystem being built at the bottom. Uniquely, *CDBakeOven* offers two modes for creating data tracks: a backup mode and a custom layout mode. The latter gives you free reign to create whatever filesystem structure you want, while the backup mode provides a simpler interface for

mirroring files and directories from your hard drive.

CDBakeoven's interface has some other nice touches, including full support for drag-and-drop, a filter tool in the source directory list (to list only matching files), and the ability to keep a running total of the size of the CD's contents as you build it. Another handy feature is the Properties Editor in the ISO filesystem lister. When burning hybrid ISO/HFS or CDs with Joliet extensions, this lets you choose which filesystems a file or directory will be accessible from.

Linux Format VERDICT

Installation	8/10
Documentation	2/10
Features	7/10
Ease of use	8/10

Not finished, but still very functional for mastering and writing ISO images. Stability and lack of audio support are problematic, but this is one to watch.

Linux Format RATING

6/10

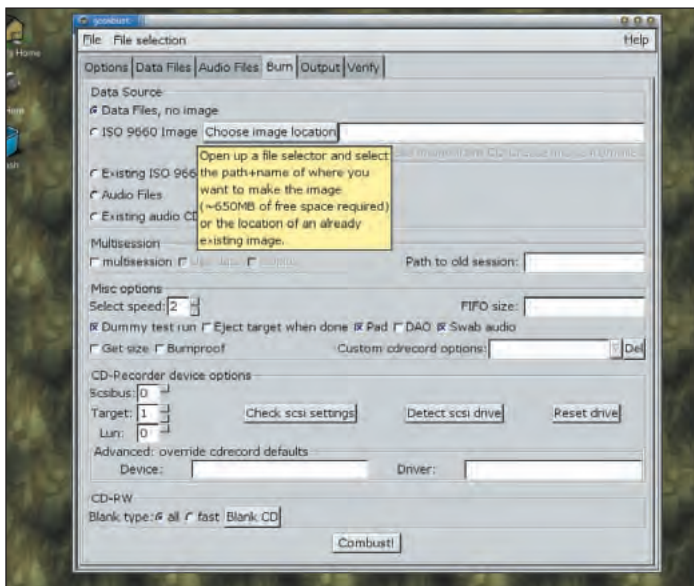
GCombust

■ Version 0.148 ■ WEB <http://www.abo.fi/~jmunsin/gcombust/>

GCombust is a **GTK+-based** application for mastering audio and data CDs. Unfortunately, its user interface is so badly designed that for

anything other than simple tasks it is just too much hard work.

A point in *GCombust's* favour is that it does allow quick access to



GCombust is functional but its interface is confused and cluttered.

functions such as writing ISO images to disc, generating ISO images from CDROMs and copying CDs, but it's only real saving grace is that it provides an interface to *cdlabelgen* to generate, preview and print inlays for CD cases.

GCombust's main window is split into six tab panes. The first lets you pick ISO filesystem options and is straightforward. The second is where you construct the filesystem for a data track, and this is where things start to fall apart. This presents a flat list of the filepaths that will be passed to *mkisofs*, and you can add and remove files or directories here with the buttons at the top. For some reason, though, you must be careful how you select directories in the chooser that pops up. You can't enter a directory and hit 'OK'; you must select it from its parent. Luckily, you can also drag-and-drop files and directories from a browser such as *Nautilus* or *Konqueror*. When you add a directory to *GCombust's* lister, you cannot browse its contents there. You can, however, select which files will be included in a particular directory by attaching patterns to match using the controls at the bottom. The third pane of the interface is used for mastering audio and mixed mode CDs and lets you add WAV files or ISO images as CD tracks.

To actually burn a CD, you need to select the fourth pane, and here the interface gets really messy. The first problem is selecting the data source for the CD. Do you want to burn on the fly from a filesystem you just created on the data track pane or write it to an image file? Or do you want to write an audio or mixed mode CD from the table of contents created with the audio pane? Mixed in here are functions to select an existing ISO image, CDROM or audio CD as a data source and the author has crammed in various burning options, the CD-RW erasing tools, and yet another set of controls to choose the output device!

Linux Format VERDICT

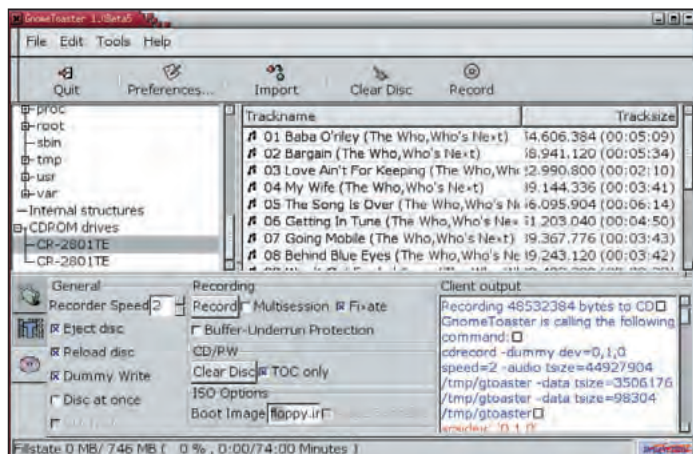
Installation	8/10
Documentation	5/10
Features	6/10
Ease of use	5/10

Offers all the basic features you need for reading and writing CDs but because of its dire user interface has little to recommend it.

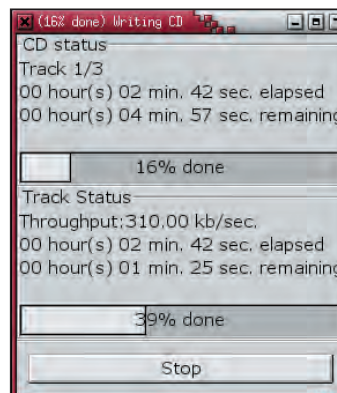
Linux Format RATING

5/10

RoundupCD Burning



GnomeToaster's split-pane interface is flexible but difficult to navigate.



This is GnomeToaster in action, burning a CD-ROM.

GnomeToaster

■ **Version** 1.0 Beta 5 ■ **WEB** <http://gnometoaster.rulez.org/>

When I first loaded up

GnomeToaster, I took one look and thought "That's hideous." Luckily, I persevered past its non-intuitive interface, lack of documentation, and byzantine configuration screens to uncover what is actually quite a good CD writing application.

GnomeToaster's forte is audio work. It can rip tracks from CD, encode them (by default) as raw, WAV, shorten, MP3 or Ogg Vorbis files and write any of these formats on-the-fly as audio tracks. CDDA look-ups are supported for retrieving track listings via the Internet, and *GToaster* can read and write CD-Text – although, as is usual, writing is only supported in DAO mode. The various audio formats it understands are all encoded and decoded via external tools – such as *sox*, *mpg123*, *ogg123* and *shntool* – and, of course, these must be installed to handle the relevant file types. The preferences dialog lets you redefine what tools handle which file types and lets you define additional file types. *GToaster* also lets you play audio tracks from CD or from files – either by double-clicking or by dragging-and-dropping them on the play button hidden away in the bottom right of the main window.

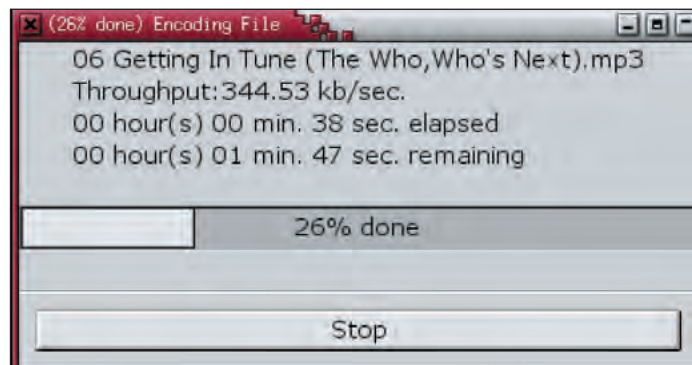
Getting the drop

The whole application, in fact, relies on the drag-and-drop technique to get things done. Unfortunately, though, its never clear until you've tried it quite what you can drag, where you can drop, and what the result will be. This is symptomatic of what's wrong with

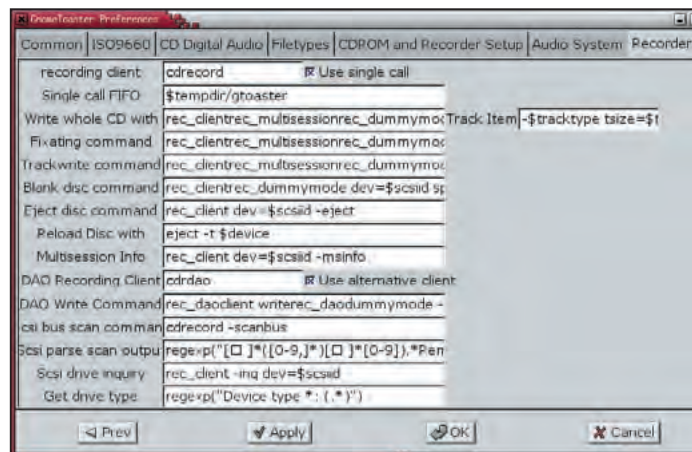
the *GToaster* user interface. But although it can take some time to find your way around in *GToaster*, once you do you'll find that it's very quick to use.

One other significant problem with the GUI is with its settings dialog. Due to the vast number of configuration

options, *GToaster* is incredibly flexible, but this will be just bewildering to new users. Thankfully you shouldn't need to change things very often and the default settings should suit most. In the most part, editing *GToaster's* configuration consists of modifying the command-line strings that are actually used to call the various utilities – *mkisofs*, *cdrrecord*, *cdda2wav* and so on – which it uses to do its job. Some aspects of the preferences are more transparently implemented, however, such as the automatic probing for



GnomeToaster excels at audio work, particularly ripping and encoding.



The configuration dialogue is not for the faint-hearted.

SCSI generic devices.

The main window of *GToaster* goes for the familiar split-pane approach. The top half is for selecting data sources and the bottom for building the CD's contents. The top left pane displays a filesystem tree containing possible data sources – these include the file system on your hard drive, the audio or data tracks on any currently inserted CDs and a 'virtual' empty ISO filesystem. You add files or tracks to the CD by dragging items from here to the bottom pane. The empty ISO object is used to create a new blank data track on the CD.

The bottom pane of the interface is further split into three tab panes. The first shows the contents of the ISO filesystem being built; you can drop files and directories here to add them, create new directories and rename objects to build the tree. The second tab pane show a listing of audio and data tracks on the CD being mastered; you can drop audio files or ISO images here to add them as tracks. The last pane contains various controls for actually burning the CD to disc and also provides some options to configure CD writing, to select a boot image and to erase a CD-RW. Having these here is actually less useful than it seems, though, because some frequently used options – such as whether to cache audio and data tracks on disk before writing or burn them on-the-fly – are buried away in the preferences dialog, while the main window provides controls for setting options which you'll hardly ever change, like the CD writing speed and whether to turn on BurnProof support.

The shortcomings of *GToaster's* interface are not a problem once you learn your way around, but with a GUI re-think it could be a vastly improved application. My wish list would include better controls for navigation, the ability to re-order tracks on the CD you are mastering and a one-click CD copy function.

Linux Format VERDICT

Installation	7/10
Documentation	2/10
Features	9/10
Value for money	6/10

Don't let the non-intuitive interface put you off. This is easily the most able CD writing app on test and trounces the rest for audio handling abilities.

Linux Format RATING

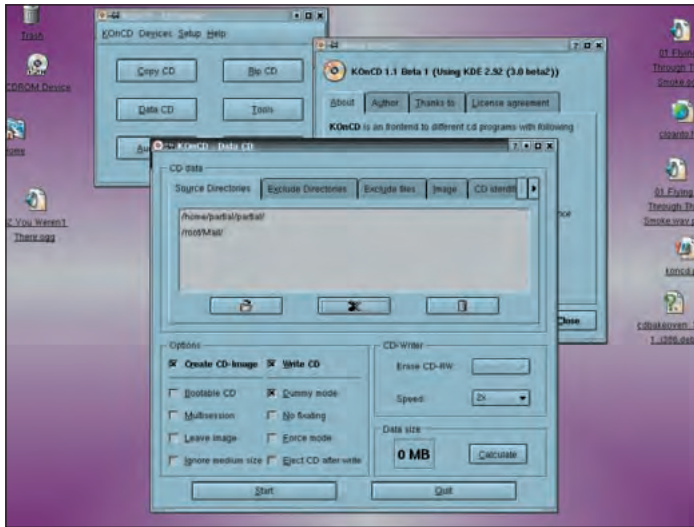
8/10

KOnCD

■ Version 1.1 beta ■ WEB <http://www.koncd.org/>

KOnCD is relative newcomer to the CD writing market on Linux, but I suspect it is one that is going to

become very popular soon because it is due to ship with KDE 3.0. It's currently in beta status, however, and



KonCD's tools for building ISO images are woefully primitive.

much less than stable. Of course, these could be due to unsquashed bugs in the beta release of KDE 3.0 rather than the KOnCD itself, but, anyway, KOnCD is not at all reliable yet, especially when handling audio.

Despite these problems, KOnCD is well worth a test drive, if only because it does have a couple of really neat features. For one, it integrates well with the KDE desktop. You can drag and drop files to add them to an ISO filesystem, for example. But the highlight for me is that KOnCD uses KDE's cdaudio I/O slave to rip tracks from audio CDs. This includes a Cddb-look up facility to rip and encode tracks in WAV, MP3 or Ogg Vorbis formats. Additionally, KOnCD supports both the reading and writing of track listings as CD-Text.

KOnCD has one of the better designed interfaces of all the applications on test here. It is simple, mostly intuitive and the controls fall readily to hand. KOnCD is functionally less capable than many of its rivals, though, and one area in which it is found to be lacking is in mastering data tracks. KOnCD's interface here simply mirrors how file paths are passed to mkisofs. It provides three

listviews with which you construct the directory tree to be stored on the data track. The first contains the files and directories to add to the image, the second lists directories from these to exclude and the third lists files to exclude. This is woefully primitive and not at all obvious unless you are familiar with mkisofs. KOnCD doesn't even provide the capability to rename the files and directories that you add – they receive the same path that they have on disk – so, KOnCD in its current state is really only useful for archival purposes.

Linux Format VERDICT

Installation	8/10
Documentation	7/10
Features	7/10
Ease of use	8/10

KOnCD shows some promise, but the tools for mastering data tracks are crude and stability is a real problem.

Linux Format RATING

5/10

KreateCD

■ Version 1.1.0 ■ WEB <http://www.kreatecd.de/>

KreateCD is a KDE-based CD writing app that dares to be different. Rather than going for a multi-paned user interface like all the others on test, KreateCD takes a multi-windowed approach. Personally, I love this and find KreateCD's GUI to be much more elegant and flexible than most of its rivals. (If you don't like it, though, there's an 'Advanced Mode' which apes the more traditional look and feel of the other apps.)

The main interface lists the audio and data tracks on the disc currently being mastered and provides controls to create and manipulate tracks.

Clicking a track brings up the track properties editor, and here you may select the data source for a track (that is, is it an existing data or audio track on CD, a filesystem, or a file?) and whether to cache the track on disk as an image file before writing.

With audio tracks, you can either select an existing CD track or a WAV, AU or Ogg Vorbis file as the data source. Also, when writing from one of these sound formats, the options

editor lets you edit the track's decoded waveform. Here you may edit the waveform's range and adjust its volume and balance. Only stereo CDDA-frequency waveforms can be used with KreateCD, though, because it has no support for conversion tools.

If you want to write a data track, you can choose either an existing track

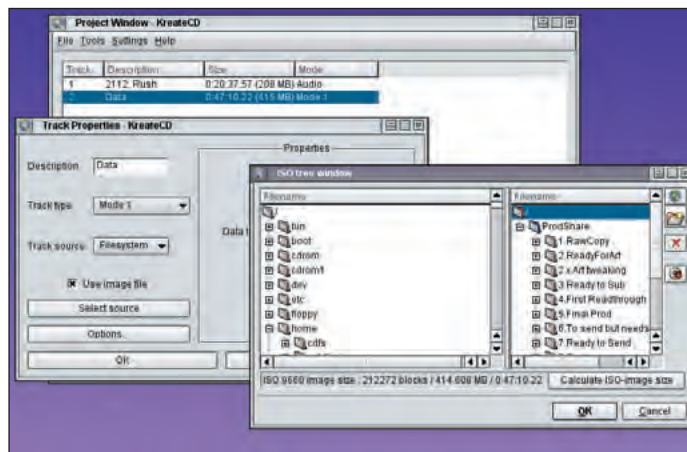
or an ISO image as a source – or create your own filesystem. The ISO tree editor is used to build the filesystem to be stored on the data track. This provides a familiar Commander-like double listview interface in which you can drag-and-drop files and directories to build your ISO tree. It's simple, direct and it works well.

In addition to its standard, general interface mode that I've just described, KreateCD can also present reduced interfaces designed for particular tasks. The first of these, the audio CD mode, is much the same except it

won't let you create data tracks; the next, the data CD mode, hides the track editor completely and just presents you with the ISO tree editor. CD Copy mode is even simpler: you merely have to click the button labelled 'Write CD'.

Another neat feature of KreateCD is that it adds an option to Konqueror's pop-up menus which lets you launch KreateCD with a directory as the source for a data track.

All in all, KreateCD is good all-round application, but it lacks too many features to be great. For example, it cannot be used to master bootable CDs and doesn't support Cddb look-ups or CD-Text.



KreateCD has an effective, multi-windowed interface.

Linux Format VERDICT

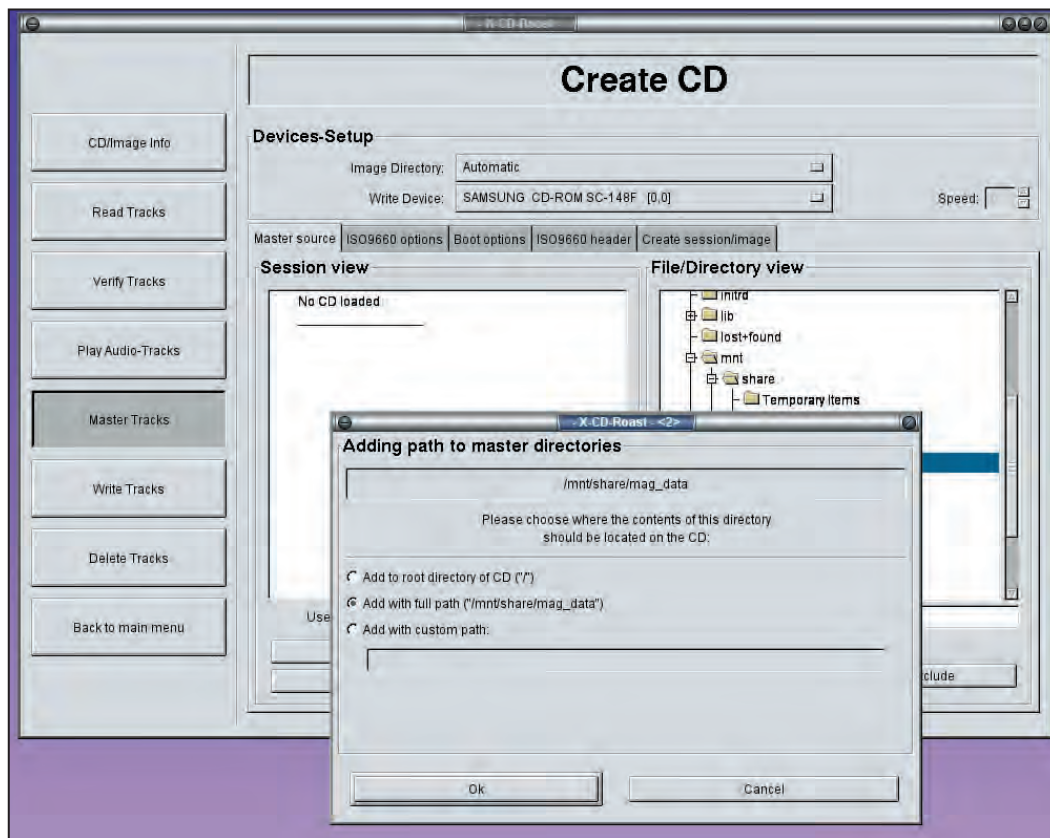
Installation	8/10
Documentation	6/10
Features	8/10
Ease of use	7/10

An elegantly designed interface and good audio capabilities can't hide KreateCD's lack of features.

Linux Format RATING

7/10

RoundupCD Burning



XCDRoast

■ Version 0.98alpha9 ■ WEB <http://www.xcdrtoast.org/>

The most popular CD burning software on Linux is probably *XCDRoast*. It's been around for ages and ships with most Linux distributions. Earlier versions sported a TCL/Tk interface, but thankfully it uses now the more modern *GTK+*. This release is still nominally in an alpha state, but it is actually rock solid – we've used it heavily here without problems. Alpha generally means not feature complete, but, despite this, *XCDRoast* offers the core functionality you need for mastering and writing CDs.

An attractive feature of *XCDRoast* is that it can be configured for non-root use. This requires correct installation, but since most distros come with *XCDRoast* pre-packaged this is not much of a problem. The initial configuration – such as selecting which drive is the CD reader and which is the burner, the default CD writing settings, the image directory for storing tracks, etc. – must be performed initially as root. This is straightforward since *XCDRoast* automatically scan for SCSI generic devices and detects CD-ROM drives and writers. The root user can then

administrate user privileges and can decide which users can use *XCDRoast* and which settings they can change.

XCDRoast provides two basic modes of operation 'Duplicate CD' and 'Create CD'. Both use the same main interface, but the former provides a reduced set of controls geared towards copying all of the tracks from an existing CD-ROM (this can be done either on-the-fly or an image of the CD can be written to disk). The main GUI for both modes is split into several pages which are accessed via a strip of buttons down the left-hand side of the main window. These are 'CD Info', 'Read Tracks', 'Verify Tracks', 'Play Audio Tracks', 'Write Tracks', and 'Delete



XCDRoast is the dependable work horse of CD writing.

XCDRoast is well-equipped for mastering data tracks.

Tracks'. In addition, when creating CDs, you also get a 'Master Tracks' page where you can master data tracks from your filesystem. On the whole, the *XCDRoast* interface is clumsy but easy to use, and the pop-up help aids the beginner immeasurably.

Two in one

The first page of the main interface, 'CD Info', provides a table of contents for the disc currently in the CD reader and another listing any tracks that are stored in the image directory on your hard drive. *XCDRoast* can do CDDb look-ups to fetch track titles for audio tracks via the Internet and a button is provided here to do this manually. *XCDRoast* doesn't cache CDDb information and doesn't support fetching CDDb entries via HTTP, so won't work behind a firewall.

The 'Read CD' page lets you read audio and data tracks from CD and write them to your hard drive. When in 'Create CD' mode, you can select which tracks to read, but in 'Duplicate CD' mode you can only write all or none. Data tracks are stored on disk as ISO images, while audio tracks are

stored as WAV files. The 'Verify CD' page then lets you compare tracks on CD with those stored on your drive to check for accuracy. *XCDRoast* rips audio tracks with *cdda2wav*, which doesn't offer the error correction facilities of *cdparanoia*. Audio tracks stored on your hard drive can be played in the 'Play Audio Tracks' page, but there's no option to play tracks directly from CD. If you wish to write audio tracks not generated by CDDA, you can place WAV files in your image directory and *XCDRoast* will pick them up. But *XCDRoast* provides no tools for manipulating or converting audio tracks, so these must be stereo CDDA-frequency WAVs only. No other format is supported.

You can create your own data tracks in the 'Master CD' page, and this is one of the best-implemented aspects of *XCDRoast*. Here you can build a directory tree with files and directories from your hard drive and either write it to an image file or directly to CDROM. Most of *mkisofs*'s features are supported – although there's no multi-session support yet.

If you don't wish to write on-the-fly, the 'Write Tracks' page lets you write audio and data tracks from your hard drive to CDROM. In 'Create CD' mode, the GUI is rather convoluted. First you must go to the 'Layout tracks' tab and pick which tracks you want to add to the table of contents. You then hit 'Accept layout' and you will be taken to the 'Write tracks' tab, which provides the controls to write the tracks to disc. (Of course when in Copy CD mode, you only see the 'Write tracks' tab, since the table of contents of the source CD is duplicated.) An annoying aspect of *XCDRoast* is that if you then go to another page – for instance, the 'Play tracks' page to check an audio track – it forgets your CD layout and you'll have to construct it again.

Linux Format VERDICT

Installation	8/10
Documentation	5/10
Features	7/10
Ease of use	7/10

XCDRoast provides the basic tools for ripping and burnings CDs and, although poorly-equipped for audio work, is one of the best packages available for mastering data tracks. The interface is poorly thought-out in some areas.

Linux Format RATING

7/10

CD burning **The verdict**

The question is, then, which of these seven CD writing applications on test here is the best. Well, as usual, it depends very much on your preferences and needs.

They all share a common functionality, namely the ability to master and write data and audio CDs. The level of frills above this core functionality varies greatly from application to application, though. Most add support for multi-session writing and creating bootable CDs (El Torito); some, like *GnomeToaster*,

enhance audio support with functions to rip audio tracks from CD and encode them to WAV or MP3 files and to burn on the fly from common encodings; others add time saving features such as support for CDDb look-ups to retrieve tracks listing from the Internet and to write read and write track listings as CD-Text.

Interface style and ease-of-use varies between the applications, too – perhaps more than you would expect from such functionally similar pieces of software. For the most part, however, they could all use significant work in

the user-interface department.

GCombus should be singled out for the booby prize, though, since its interface is just dreadful.

The last crucial factor is stability. Several of these applications still have a beta status, so a less than perfect robustness can be forgiven; *XCDRoast* and *GnomeToaster*, however, seem to be stuck in a perpetual state of beta testing, although in *XCDRoast*'s case this is not a problem, since it is actually a solid as a rock.

And the winner is...

If I had to choose a best all-round CD-writing application, then – much to my

surprise – I would have to choose *GnomeToaster*. At the beginning of this test I didn't think it would be receiving high marks – but, despite its non-intuitive interface and complete absence of documentation, it boasts the best all-round feature support. It is also, with the ability to rip to and burn on-the-fly from an extensible range of audio encodings and with good support for CDDb and CD-Text, the handiest tool for working with audio tracks. The developers needn't be that smug, however, because *GToaster*'s GUI badly needs an overhaul to improve navigability and general ease-of-use. In fact, the preferences dialog should be taken out and shot now, and any kind of documentation at all would be an improvement.

As a runner up, I would like to nominate *KreateCD*. The lack of support for CDDb, CD-Text and mastering bootable CDs are all points against it, but its flexible and consistent user interface make it a joy to use – even though the interface could do with more help. It also boasts several unique features, such as its minimal waveform editor and its ability to integrate highly with the KDE desktop.

At the top of the up-and-coming list you would have to find *KOnCD*. It's barely usable at the moment, but it has the makings of an incredibly slick interface and feature support is already looking good. 

Table of features

	CDBakeOven	GCombus	GToaster	KonCD
Version	1.8	0.1.48	1.0beta5	1.1beta1
Interface	KDE	GTK+	GNOME	KDE
Multisession	Yes	Yes	Yes	Yes
Burnproof	Yes	Yes	No ²	Yes
Erase CDRW	Yes	Yes	Yes	Yes
Write multimode discs	Yes	Yes	Yes	No
Write on the fly	Yes	Yes	Yes	Yes ⁴
Rip audio tracks	Yes ¹	No	Yes	Yes
Play audio tracks	Yes ¹	No	Yes	No
Supported formats	WAV	N/A	WAV, SHN, MP3, Ogg ³	WAV, MP3, Ogg
CDDb lookup	No	No	Yes	Yes
CD-TEXT	No	No	Yes	Yes
El Torito	Yes	Yes	Yes	Yes
Hybrid/HFS	Yes	Yes	No ²	No
Rating	6	5	8	5

	KreateCD	XCDRoast	webCDwriter
Version	1.1.0	0.9.8 alpha9	2.2
Interface	KDE	GTK+	Java (Swing)
Multisession	Yes	No	Yes
Burnproof	No	Yes	No
Erase CDRW	Yes	Yes	Yes
Write multimode discs	Yes	Yes	No
Write on the fly	Yes	Yes ⁴	Yes
Rip audio tracks	Yes	Yes	No
Play audio tracks	No	Yes	No
Supported formats	Wav, Au, Ogg	WAV	WAV, MP3
CDDb lookup	No	Yes	No
CD-Text	No	Read	No
El Torrito	No	Yes	Yes
Hybrid ISO/HFS	No	No	Yes
Rating	7	7	8

1 Feature unimplemented or broken in test version.

2 Feature could be supported if the command line calls for *mkisofs/cdrecord* were re-written in the settings.

3 Default options. Audio file format support is extensible.

4 Data tracks only

Web links

CD Writing HOWTO

<http://www.linuxdoc.org/HOWTO/CD-Writing-HOWTO.html>

Parallel port ATAPI CD-R information

<http://www.torque.net/parport/cdr.html>

cdrtols (cdrecord, cdda2wav, mkisofs)

<http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html>

cdrdao

<http://cdrdao.sourceforge.net/download.html>

cdparanoia

<http://www.xiph.org/paranoia/>

cdlabelgen

<http://www.red-bean.com/~bwf/software/cdlabelgen/>

HotPicks

The best new open source software on the planet!



Maurice Kelly

Busy coder,
electronic
engineer and
Midnight Oil fan

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 CD, but we've provided web links if you want to make sure you have the very latest version.

If you have any suggestions for things that we should cover, email us at linuxformat@futurenet.co.uk

HotPicks at a glance

Gnumeric	46
Endeavour Mark II	47
Trommler	47
KDevelop	48
LTris	49
GLAME	49
SCREEN	50

HotPicks award

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!



TOP SPREADSHEET

Gnumeric

■ **WEB** <http://www.gnumeric.org/> ■ **VERSION** 1.0.2

Gnumeric has been one of the flagship applications of the GNOME project for a long time, despite the fact that it has only recently gone into its 1.0 release phase. In case you're not aware, *Gnumeric* is a spreadsheet application which aims to be a complete solution for those who are dependent on using *Excel* under Windows.

Unlike *Excel* and many other spreadsheets, *Gnumeric* has rather a basic look and feel to it. There isn't a whole lot going on in the way of toolbars and extras but this should be seen more as a benefit than a loss. *Gnumeric* is a straightforward application that has more hidden depth which can be extracted when necessary. It contains a massive number of functions that can be used

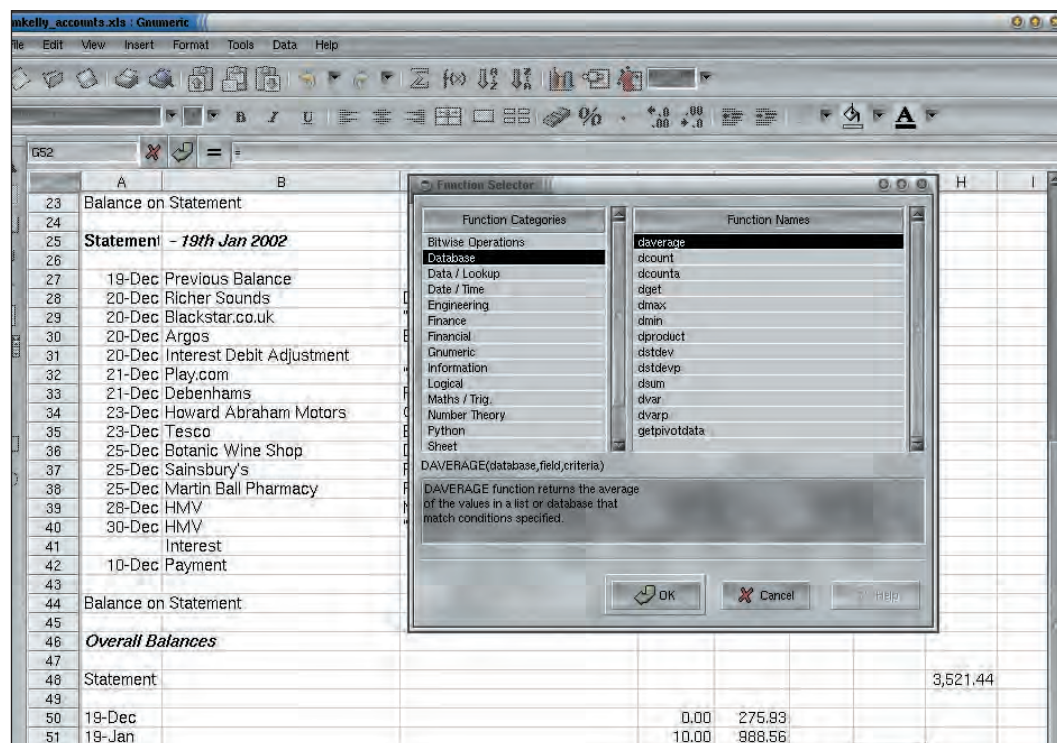
in the creation of complex spreadsheets for a wide variety of purposes. Functions can be typed directly into a cell, but can also be conveniently chosen from the Function Selector, which breaks down the list into categories with a concise explanation for each option.

A number of data analysis tools are also available including analysis of variance and Fourier analysis. If you install the *Guppi* graphing tool (<http://www.gnome.org/projects/guppi/>) it is possible to display complex graphs. For those concerned with presentation of data there are a wide range of cell formatting options, from numeric types to colours and shading. Spreadsheets of considerable size can be handled - eventually I got tired scrolling down and across, but loading

up some of my old *Excel* workbooks yielded thousands of rows, 100 columns and at least 40 worksheets.

Stability-wise this application cannot really be faulted - older versions were robust but not quite as stable as this. In general use the program runs very quickly - load times are minimal and recalculating large spreadsheets did not take excessive amounts of time.

Of course, one of the biggest factors people consider when choosing to use a particular spreadsheet is compatibility with *Excel*. *Gnumeric* uses its own native XML-based file-format, but it can read (and to a lesser extent write to) the *Excel* file formats. The feature list for *Gnumeric* states that it supports 95% of the *Excel* function list, but with such large numbers of functions it is obviously hard to verify this. It is safe to say though that there will be few people who won't find their *Excel* compatibility needs fulfilled by *Gnumeric*. There is now one less reason for people to continue to use Windows.



Gnumeric is fast becoming a darn good reason to leave Windows behind.

SPEEDY FILE MANAGER

Endeavour Mark II

■ **WEB** <http://wolfpack.twu.net/Endeavour2/> ■ **VERSION** 2.1.4

With the development of *Konqueror* and *Nautilus*, file-system browsers for Linux are becoming plentiful. The problem with the *KDE* and *GNOME* offerings is that they can be quite hefty, and desire to do things like take over your desktop and connect you to Web and FTP sites. There is a market out there for plain old file browsers, and that is where *Endeavour Mark II* comes in.

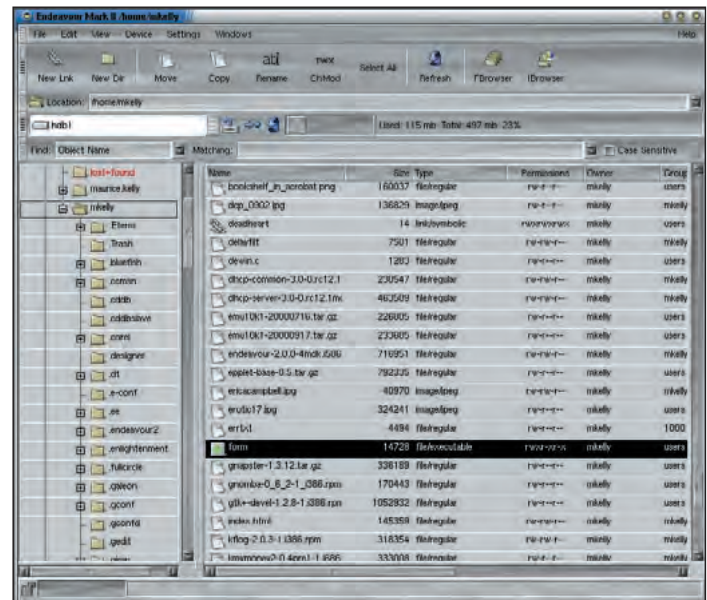
Endeavour is a plain old file-browser along the lines of *Windows Explorer*. Toolbar at the top, folder hierarchy to the left-hand side, and folder contents on the main pane to the right. So far, so bog standard. But what makes *Endeavour* that little bit nicer than most of the competing applications out there is that it gets the job done, and it does it quickly. Instead of previewing document contents, and producing nice thumbnails of images on the fly, *Endeavour* just gives you the facts.

As you would expect from a file browser, double clicking on a file will

cause it to open, although you will need to edit your MIME types to set up all but the most basic of file associations. Thankfully the system for adding and editing MIME types is fairly intuitive and there is a brief example in the online documentation.

Whilst *Endeavour* doesn't have the overhead of in-line image thumbnails, it does come with a neat, *Imlib*-based, image browser. This can be kicked open when necessary and consists of a simple layout – folders, files and thumbnails on the left, and full image viewing on the right. Like the rest of the application, this mini-app is responsive and does the job with a minimum of fuss. Some of the other projects out there should take note!

As well as standard file and image browsing, *Endeavour* features a handy recycle bin. As you would expect, deleting a file within the browser plops it into the recycle bin from which it can be restored at a later date. Not one of life's essentials, but useful to have. Throw in the ability to mount and unmount filesystems and you end up



Despite looking a lot like *Windows Explorer*, *Endeavour* is not going to pull down the entire operating system in a hurry.

with a great application that offers an alternative to the command line,

without the overhead of a more fully featured file browser.

DRUM MACHINE

Trommler

■ **WEB** <http://www.muth.org/Robert/Trommler/> ■ **VERSION** 3.0

To go along with the great *GLAME* package (see page 49) we thought you might want to continue the musical theme with this neat little drum machine. *Trommler* is what can only be described as a compact application – everything happens in one window. It is *GTK+*-based so you'll need to have your *GNOME* up-to-date in order to use it.

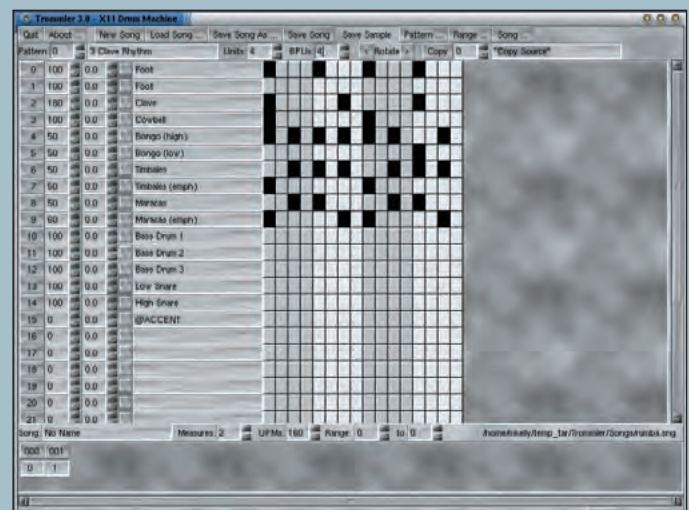
Trommler is distributed as a package containing the source and a pre-built executable. While it is tempting to execute it straight away you'll need to set up an environment variable as described in the README – either that, or sit about wondering why there's no sound coming from your speakers.

It comes with a sample file called *rumba.sng* (located in the

Songs directory just off the main source.) It's easiest to load this file up and start playing with it to get a feel for how the application works. Down the screen you should see a list of drum types. Across the screen is the timeline for each of those drums – you instruct *Trommler* to play a drum at a particular time by clicking the mouse at a point along the line. That point will turn black to indicate that a note is being at that time.

Once you have defined your pattern, you'll need to set the "Play" radio button at the bottom of the screen to "Song." This will play the entire piece repeatedly so that you can hear it as a continuous rhythm. Caution is advised – it can become very tedious!

With the possibility of up to 31 different drums being utilised at any

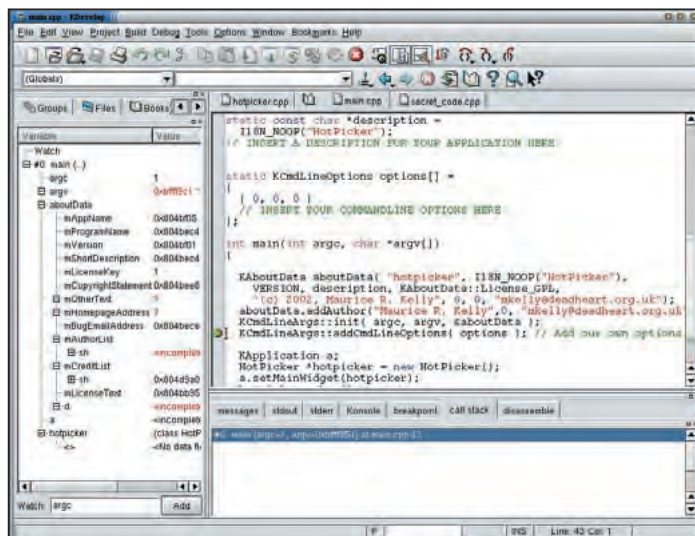


Like all good UNIX programs should, *Trommler* serves a single purpose and serves it well.

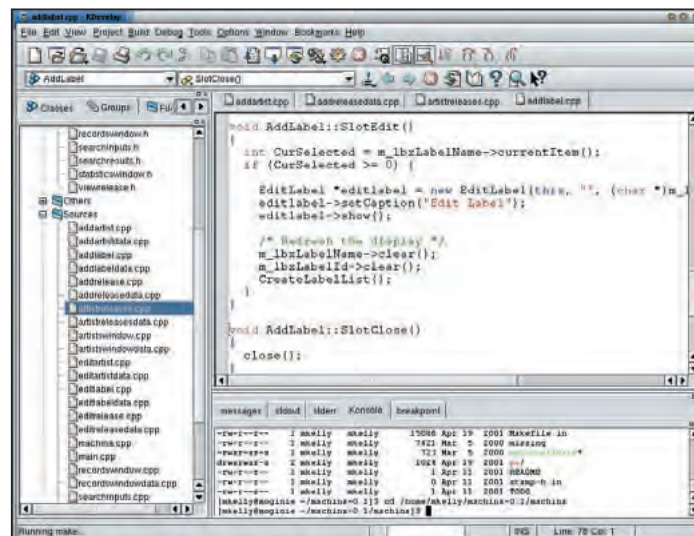
one time and 128 patterns to use, there is an amazing range of possibilities in this application. Once you've created your very own drum masterpiece you can save it out as a sample.

Trommler is a very simple application – it has loads of functions

to tweak and adjust, and it is very easy to waste hours of your time messing about to get that one great rhythm. The only downside is that you don't get enough of a time-frame to lay down a cracking drum solo. Back to the pots and pans for that one.



Of course, we all know that the best coders don't ever need debuggers, but *KDevelop* threw one in for the rest of you.



Managing large software projects becomes a lot easier with *KDevelop*, especially when it comes to maintaining a large number of classes.

KDE'S CUSTOMISABLE C/C++ IDE

KDevelop

■ WEB <http://www.kdevelop.org/> ■ VERSION 2.0.1

Software millionaires – there are not as many of them crawling out of the woodwork these days now that the dotcom bubble has burst, but that doesn't mean there's no room for a bright spark with a genius of an idea. There's still money to be had and if you can come up with a killer application, then maybe you can be the one to get your greedy mitts on it. So you've got a great software idea for that killer app – what do you do next? Get *KDevelop* and get the code written!

KDevelop is one of the best Integrated Development Environments (IDEs) available for Linux and, as you have probably guessed, it is completely free. You'll need the *KDE/Qt* libraries to run it (not a problem for most recent distributions) and the development libraries (including the sources) in order to actually develop anything useful.

At first glance, it may seem like a text editor with a filesystem browser tucked into a pane on the left hand side, but there's really a lot more to it than that. *KDevelop* will gladly help you manage your project – you don't have to worry about those pesky Makefiles and other things that have hindered software developers for aeons. If you decide to add a few more source files, *KDevelop* handles the Makefile updates – you just have

to keep pressing the Execute... button to see how it's looking! Controlling the build options is as simple as ticking a few boxes in a dialog, and creating yourself a new class and associated source files is a breeze.

To get started with *KDevelop* it's best to kick off a new project – the Application Wizard is tucked into the Project->New... menu item. At this stage you will want to decide exactly what type of application you are going to create. It's easy to get confused about *KDevelop* as a result of the name – it isn't just for creating *KDE* applications. Granted, it does focus on (and is heavily biased towards) *KDE* development, but when presented with the Application Wizard you can appreciate the work that has gone into this suite. As well as numerous types of *KDE* applications, you can also start a project for *GNOME*, pure *Qt* (the underlying toolkit behind *KDE*) and even console applications.

Once the application type is chosen, it's a simple case of filling in project details – well, there is the hard bit of choosing a name – defining project directories, and fun stuff like picking your desktop icons. Once you've clicked through a couple more screens (or modified them if you want to edit your templates and version control system information) you get to

press the Create button. At this point *KDevelop* generates all the required project files – the Makefiles and the nice directory structure and so on – and once that's done you're ready to press the Exit button and get stuck in to your code. Ah, forgot to mention the hard bit – actually writing the software...

Assuming you know how to program in C++ (or C) you should have no problem diving into the newly generated source files. It's worth getting a tutorial to give a bit of guidance on writing *KDE* applications, but otherwise it shouldn't be much of a challenge. You get a nice class browser in the left hand pane, a handy file grouping view (collecting headers into one folder, sources into another, and so on) as well as a standard file browser. In the right-hand pane you can view your source files, as well as help documentation.

The editor is extremely competent, featuring a wide range of customisation options (auto-indent, auto-brackets, etc.) and a neat syntax highlighting implementation, again with lots of control. Below the editor is the message window containing five tabs for message output and breakpoint monitoring during debugging sessions. It even has a built in *Konsole* offering quick access to the command line.

KDevelop has support for running external debuggers, but also has it's own inbuilt system for stepping into the internals of your code when it's not doing quite what you wanted. Breakpoints can be set in the main editor, and a stack monitor and disassembler are presented in the messages window. You can also set

watches on particular variables and class members.

It is possible to create your user interfaces by hand, but it is much easier to utilise the *Qt2 Designer* tool from TrollTech. This is a third party tool which you can invoke from *KDevelop*, but it really makes interface design and implementation a lot easier. As well as the tools mentioned above, *KDevelop* also boasts the ability to execute user-defined extra tools, and to create and manage documentation for your project, which makes it an even better environment to foster your latest killer app.

However, the difference between a glorified text editor and compiler combo, and a really great IDE, is the support it gives the developer. *KDE* provides plenty of support by keeping all the major tools accessible from within one interface, but also gives the developer an extensive range of information at their fingertips. The amount of info is somewhat dependent on what you have installed on your system, but if you have the right stuff you get a well-presented index to class documentation, *KDevelop* manuals and general *KDE* documentation. If you have a search system such as *ht://Dig* installed you can utilise it to index the documentation to make it easier to find what you're looking for.

If you are a C/C++ developer under Linux, and you feel the need to write software using a graphical user interface, then you'll probably want to check out *KDevelop*. It's not quite as easy as a RAD package such as *Kylix*, but it's certainly no slouch when it comes to speeding up the development process.

MULTIPLAYER TETRIS CLONE

LTris

■ **WEB** <http://games.sourceforge.net/> ■ **VERSION** 1.0.1

Even if you didn't have one yourself, everyone had a go at a Gameboy at some stage in their life (within a certain age bracket anyway.) Nintendo had the amazing sense to pack the absolutely fantastic puzzle game *Tetris* with the Gameboy, and the whole thing was a massive hit.

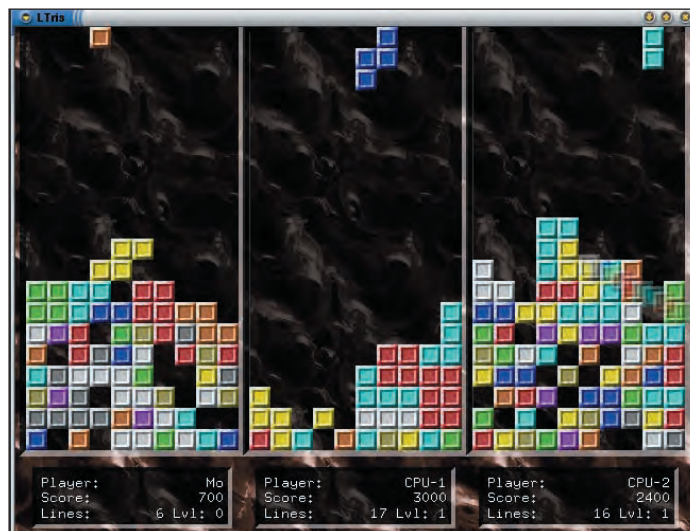
However, for those who did miss out – different shaped blocks falling down the screen must be packed as quickly as possible lest the blocks reach the top of the screen. When a complete horizontal line of blocks is achieved it disappears shoving everything else down the screen and putting you one less line in danger.

Tetris made it to just about every format under the sun (including mobile phones) including Linux, so

why is this version being declared a HotPick? The answer is simple – it is a fantastic rendition of a timeless classic. It's smooth, looks fantastic and throws in not only a two-player mode, but allows a third person to join in if you can all fit around the keyboard.

You can also engage the computer in both two and three player modes. If this isn't enough there is the additional game mode known as "Figures" – each level starts off with a bunch of blocks already at the bottom of the screen to make it more challenging when the rest start descending.

There is a great range of options available to control the game – playing screen layout, the actions of the blocks as they move down the screen (anybody else prefer line-by-line



Obviously the CPU players were using some sort of hidden cheat mode.

advancement to smoother scrolling?) and sound and graphics options. These really are fantastic, and the game looks really satisfying running in full-screen mode.

This game requires a warning – be

prepared to witness your productivity decrease drastically as a result of installing this. If they add network play there could be some serious time-wasting to be done without having to crowd people round one keyboard.

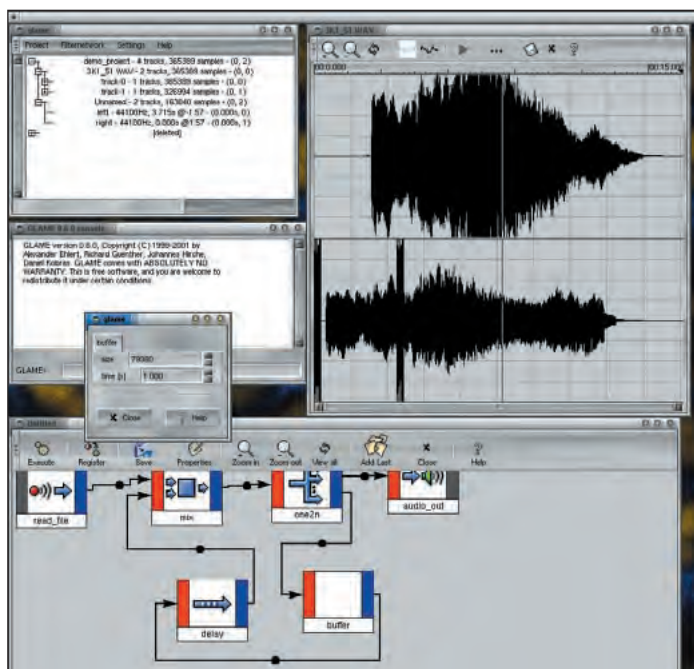
AUDIO MANIPULATION TOOL

GLAME

■ **WEB** <http://glame.sourceforge.net/> ■ **VERSION** 0.6.0

GLAME is *GNU/Linux Audio Mechanics* – a *GNOME*-based tool for serious audiophiles who

use Linux. Often described as “the *GIMP* of audio processing” Version 0.6 – the latest stable branch – has been



It's not the prettiest of applications (well it is for audio, after all), but there's not a lot you can't do with GLAME.

significantly improved in terms of user-friendliness, stability and features.

Starting *GLAME* for the first time requires you to define the location of your swapfile – if you intend to edit files which are of any significant size then it will need a lot of free space. Once done you are presented with a very bare looking interface with a few menu items along the top. *GLAME* works with the concept of projects – you create a new project (or load an existing one) and add your sound files to it. To obtain sound files you can record a file, or import a pre-recorded wave audio file. When you have some files in your project you can start to edit through the right-click menu.

Like the *GIMP* everything you do takes place in a separate window. Editing wave files is easy with the use of markers – cutting and pasting can be a bit imprecise, but is a lot easier if you zoom well into the wave pattern to get more precise positioning.

As well as performing edits directly on wave files, it is possible to use the “Filternetwork” tool to create complex, well, filtering networks. Taking input from a variety of sources it is possible to create complex filters through the large number of plugins available. Each plugin has a number of parameters which can be edited in real time to create a huge array of amazing effects.

The filter network is a very powerful tool. The bare canvas can be right clicked in order to display a list of available components which are

deposited on the canvas. Connections can be dragged between the output of one component and the input of another in order to “pipe” the audio between them. Two blocks are required for anything to happen – the input and output blocks. Connecting these directly results in an undistorted output so you need to throw some effects in the path between them. If you have experience with DSP a lot of the effects will make sense to you, but there is fun to be had in experimenting for novices and experts alike.

For experts in the scheme scripting language a console version of *GLAME* is supplied which provides direct access to the *GLAME* API & backend.

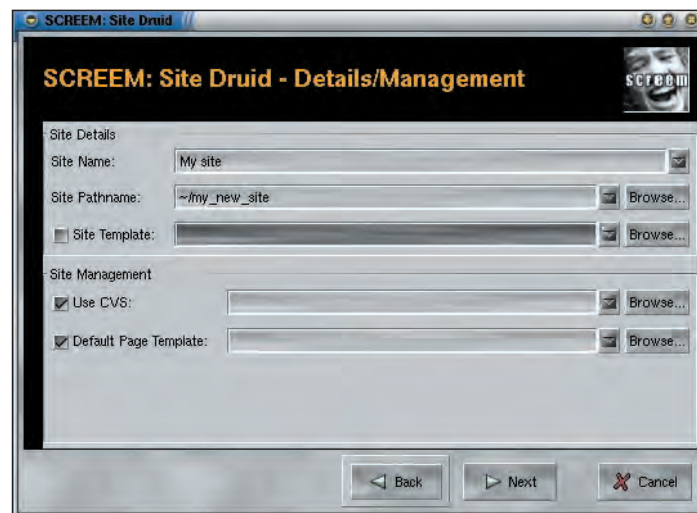
While it's possible to manipulate entire songs within the application, it may be more productive to use the audio processing power of *GLAME* to manipulate samples for use within sequencing software to give a song a more distinctive sound. *GLAME* does have its own Timeline feature which can be used to perform time-shifting of tracks relative to one another, but it is the only feature of the program which is marked as experimental and potentially dangerous.

GLAME is a delight to use, is stable and has some great documentation. If you're into audio experimentation then it is well worth checking out. In fact, it's worth getting hold of even if you just want to make your voice sound like *Alvin* and the *Chipmunks*.

HTML EDITOR

SCREAM

■ **WEB** <http://www.scream.org/> ■ **VERSION** 0.4.1



The Site Druid allows for an easy way to set up your site without having to drag in pages from all over the place.

Are you the sort of user who only uses X to facilitate the display of a large number of *xterms* on the screen at one time? The sort of user who spurns configuration tools and falls back on manually editing *rc* files for fun? Most importantly, are you the sort of user who believes that there is only one way to maintain webpages, and that One True Way is raw HTML in a text editor?

If you answered "yes" to the above questions, then you'll probably want to move onto the next page now, because *SCREAM* is not going to be for you. The *Site Creation and Editing EnvironMent* (or *SCREAM* for short) provides the rest of us with a means to edit HTML pages, and manage websites within one convenient application.

If you've ever used a program like *Homesite* under Windows, you will know roughly what to expect from *SCREAM*. The left-hand side of the window is occupied by a pane which contains such details as the directory layout, a tree view of the HTML page structure, and some properties for the currently selected tag. A complete tag listing is available which enables single click insertion into the active document. The lower portion of the screen is taken up with a status window for errors and messages, as well as a built in Guile interpreter used for user-created extensions to the application. It does take up a fair bit of screen space for what little it can be

used, so it can usefully be turned off.

The main work area defaults to the actual page editor which can handle multiple pages through the use of a series of tabs. The editor is extremely nice, featuring syntax highlighting for

HTML and PHP – though you may want to have a look at changing the colours as some are a bit harsh. The editor has a number of welcome features such as in-line tagging and the "Insert Tags" menu item.

In-line tagging results in the list of available properties for a tag being displayed after you type the tag name – e.g. if I was to type `<img` a menu appears presenting me with the option to insert properties such as **src**, **alt**, etc. The *Intellidose* feature causes an appropriate closing tag to be created once you type the `>` – it takes a bit of getting used to, but in no time you'll wonder what you did before. In a similar vein the "Insert Tags" menu item appears on the context sensitive menu – this allows you to add a property to a particular tag – the list presented only contains those properties which are valid for that tag.


One thing that *SCREAM* is not is a WYSIWYG HTML editor. It does come with a preview function to allow you to check on how your page is looking at a basic sense, but even this is not quite the same as checking it out in a real browser. Fortunately *SCREAM* can be configured to upload pages to a remote server which makes it easy to make modifications and quickly upload them for reviewing.

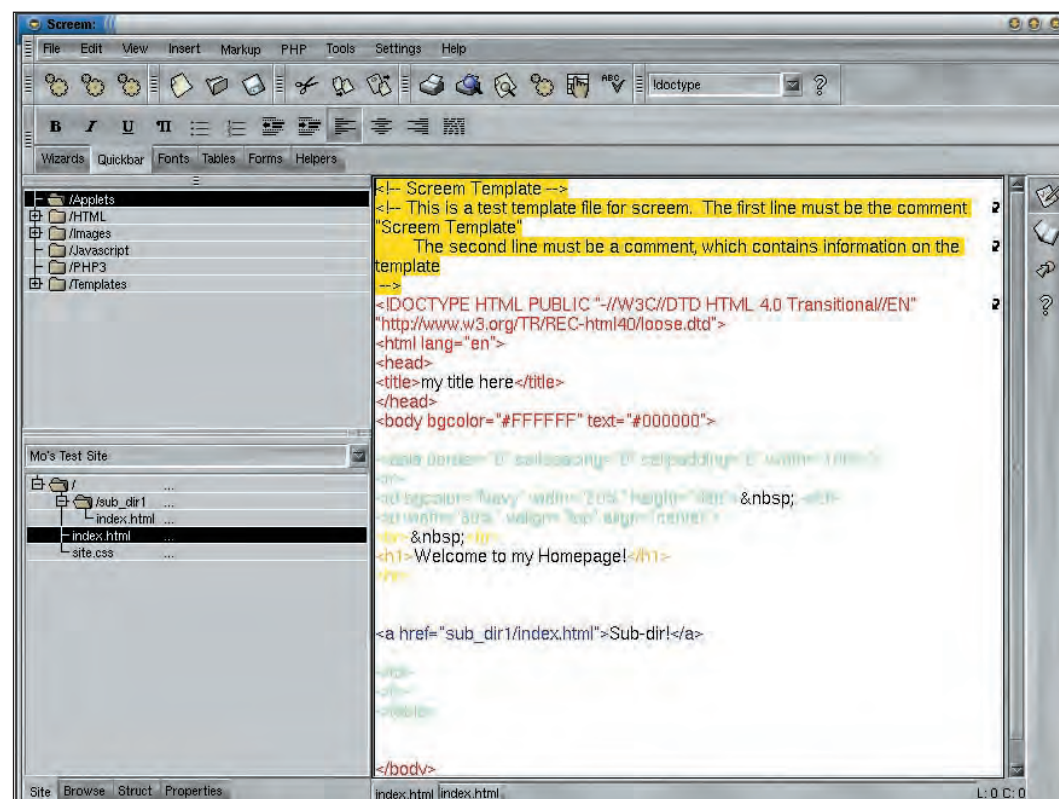
While you can use *SCREAM* to edit individual pages, most people will prefer to be able to use it to maintain

their website. You can import an existing website but it is preferential to create a new site within *SCREAM* and continue working with it from there. To do this, *SCREAM* provides us with a nice Druid for creating new sites, requiring basic information like remote server login details, site details, and template locations.

Once created the site can be managed through a dedicated pane, as well as the dedicated link view available in the main window. This is standard fare for any HTML-editing application that boasts to support more than basic sites, but *SCREAM*'s link view is a tidy little affair, indicating internal and external links, as well as those which have not been uploaded to the remote server.

If your site is being maintained by a number of people, *SCREAM* offers the option of using the Concurrent Versions System, which allows the various users to check pages in and out as they're working on them so that consistency of the pages is maintained.

SCREAM is definitely one of those programs that is in heavy development, and as such can be occasionally flaky, but it's still worth using despite some crashes. It isn't ready to start challenging the more mature HTML editor market on Windows machines, but is certainly shaping up to be a very strong competitor in the Linux scene. 



This kind of application layout will be familiar to those who are used to package like Macromedia's *Homesite*.

VoiceOverIP

Voice over IP

Will the internet now replace the need for the telephone? VOIP is a rapidly growing industry with more and more regular users each year. Here we look at how you can get into the action with Linux.

One of the early successes of Internet technology that had major corporations worried was the invention of "VOIP", or Voice over IP. Put simply, this technology allows users to encode their voice at one end of a connection and decode it at the other. Set up a system where this happens simultaneously in both directions and you have the Internet equivalent of a telephone call.

The appeal is obvious – instead of paying 30p/minute to call Uncle Bruce in Australia, you can connect

up to the Internet and pay your local access rates. Hurrah, we need never bother Busby again. Well, almost. You still need some way to let Uncle Bruce know you are calling him, so unless you arrange a predetermined time to arrange the connection, you'll still need to 'phone up to tell him to expect an Internet call.

That notwithstanding, VOIP use has been rising exponentially over the last few years, with more and more people finding it a convenient way to make not only voice calls, but also video calls and other communication

activities. Gateway services are also available in some countries, allowing you to use the Internet at your end, but route your call to a standard land-line at the other – which negates the necessity for the other person to have VOIP capability or even a computer!

The actual requirements for a VOIP terminal are fairly minimal – all you need is a computer, some sort of soundcard (full duplex), an Internet connection and some software (of which more later).

The Future

Does VOIP herald the end of the telephone? Not for a long time, and maybe not for everyday use. The problem is that whilst VOIP may be cheaper for the user, it is actually, at the moment, replacing a cheap

Telephony hardware

You might wonder why you need hardware. After all, if you have a soundcard or a webcam, you should be away. Well, not necessarily. Firstly you need a soundcard that is fully duplex. Even if yours can handle simultaneous input and output, you may find that your computer isn't up to the task of compressing the data.

A lot of 'on board' sound devices rely on the CPU a lot. When the CPU has to run compression algorithms and help out the soundcard, you may find that speech becomes interrupted and not terribly useful as a means of communication. If you currently suffer from audio dropouts during intensive CPU activity, or even while accessing drives, you'll find it will only get worse when using VOIP.

Telephony hardware is available to specifically manage the ADC and DAC operations as well as compressing sound data using the common codecs employed by VOIP. Such hardware may offer additional features, like the ability to use existing phone handsets.

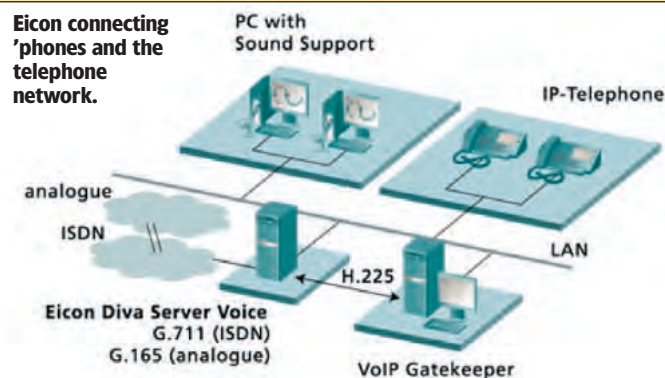
Not all of this hardware is supported by Linux, but the Quicknet Technology cards are, and have been for some time. The modules supporting them can be found in the Telephony section of the kernel, and you can find out more about the hardware at www.quicknet.net

Eicon have a range of Linux-supported cards which act as gateways between the Public Switched Telephone Network (PSTN) – i.e., plain old 'phone lines – and DSL, ISDN and T1/ethernet connections.



Specialist hardware like Eicon's Diva server range are supported by the kernel and provide interconnectivity with a variety of comms systems such as ISDN, SDL, GSM and analogue voice calls.

Eicon connecting 'phones and the telephone network.



A Linux supported webcam like this Logitech Quickcam is about all the hardware you'll need for simple video calls.

VoiceOverIP

infrastructure with an often more expensive one. The growing uptake of broadband Internet access at consumer level may make VOIP systems more viable, but currently it is used mainly by companies for tele/video conferencing.

While it is possible to achieve this on dial-up lines, you'll often find that such connections, because they are using a greater percentage of the theoretical available bandwidth, often have more problems with connection.

Where VOIP is really making inroads is with corporations who have many offices. Typically these are already connected with high bandwidth connections for using central servers, etc. Typically they all have to contact each other a lot too. The solution is to install VOIP Gateways in each location, allowing office to office calls to be made at no cost. There is a variety of hardware available for such activity, and some devices have Linux support – see the hardware boxout.

VOIP, which less than a decade ago hadn't even been contemplated, is now estimated to be a market worth \$39 billion.

How does it work?

The basic concepts behind Voice Over IP seem simple. At one end you collect the voice data, send it across the Internet to the other end, where it is transformed back into speech. There isn't really anything clever about digitising the sound (well, not any more – your soundcard has an analogue to digital converter), or about turning it back into analogue noise at the other end. The only tricky bit is trying to send it.

The problem is that the way you want the data to behave and the way the Internet treats it are two different things. Your data must be split up into packets for transmission, routed over the Internet and reassembled at the other end. The trouble is that the Internet isn't really a permanent connection in the same way that, say, a direct modem-to-modem link is.

The Internet only exists because individual packets find their own way to their destination. So your voice packets, when they arrive at the other end, could be in any order, possibly with huge segments of data arriving before other key parts. This is

rather like taking a picture, making a jigsaw out of it, shuffling them up and posting all the bits to someone in different envelopes. Imagine trying to assemble that in real time.

Fortunately, the Internet has communication standards to deal with this, called the RTP (Real-Time Transport Protocol). This doesn't solve the problem of the packets arriving at different times, but it does offer some help in reassembling the mess. One of these useful additions is to timestamp the packet – thus making it much easier to assemble the data into a meaningful order, and skip over packets that don't appear in time. The actual packets of voice and video data are communicated using UDP datagrams, which minimise the overhead.

H323

As with everything, a standard is needed so that clients can communicate with each other. The standard that emerged for VOIP is known as H323. This specifies the type of data that goes into the packets, how it is arranged, how it may be encoded and how it is handled by software. The H323 protocol goes far beyond the basic requirements for a point to point VOIP application, because it also makes provision for conference calls, bandwidth management, admission control and more.

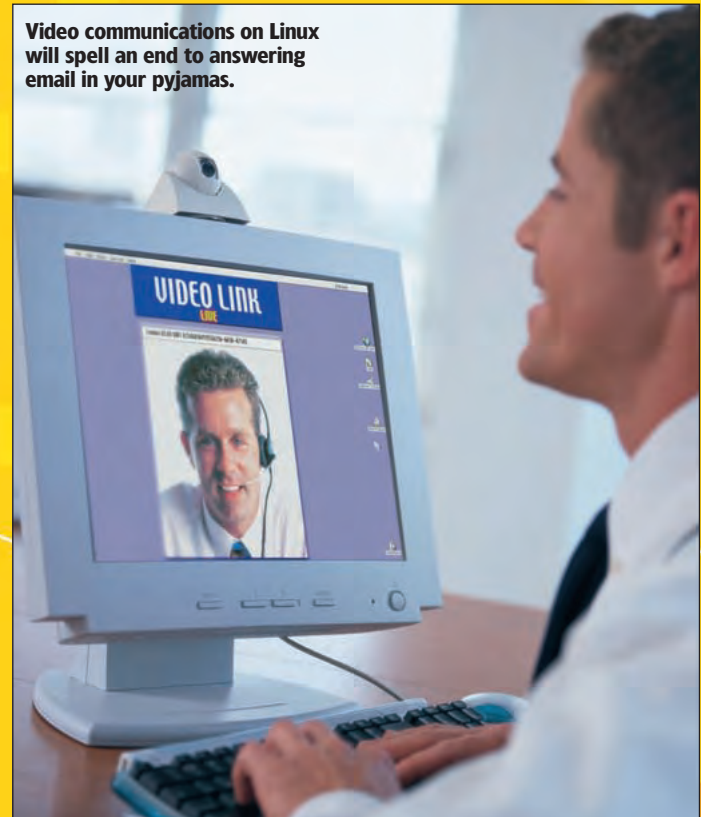
H323 defines additional devices too – to build an infrastructure to support VOIP services. As well as the clients (which are known as terminals in H323), there are Gatekeepers, Gateways and MCUs (Multipoint control units).

The gateway hardware we have already mentioned – it allows the connection of IP based calls to the normal telephone network and, occasionally, the other way round. Gatekeepers are, if you like, the digital equivalent of a super-featured telephone exchange. They can route calls, manage the bandwidth over connections and manage access (e.g., to only allow certain clients to access gateways). Gatekeepers aren't strictly necessary, but they do make more and more sense when you are considering VOIP on a larger scale. Unmanaged bandwidth can cause particular problems – the lucky client on the fast end of the network gets

to use all the bandwidth for perfect sound and video, and everyone else suffers from hiccupping audio and a picture that looks like it was being transmitted from Pluto.

Multipoint Control Units act as sort of relay broadcast stations, allowing multiple users to connect to the same stream – this is required for proper conferencing.

Video communications on Linux will spell an end to answering email in your pyjamas.



“Unmanaged bandwidth use can cause problems, everyone else suffers a picture that looks like it is coming from Pluto.”

OpenH323

The H323 standard was actually defined by the International Multimedia Teleconferencing Consortium (ITMC). This is an umbrella organisation to which many of the major telecommunications companies belong. Indeed, membership is open to anyone who wants to pay the fees (about £4000



VoiceOverIP



a year at the moment). It is a non profit organisation whose sole purpose is to help develop and promote compatible telecommunications systems for videoconferencing and the like. The ITMC provides a forum for the exchange of ideas between members and sponsors tests and trials of technologies and equipment – to help make sure everyone's devices talk to each other. They are also responsible for drafting the H323 standard and any additions or other standards that go with it.

The ITMC's main claim to fame apart from H323 (and other standards for videoconferencing over ISDN and other technologies) is that they have a formal representation to the ITU, the International Telecommunications Union. The ITU is a United Nations organisation whose purpose is, similar to the IMTC, to foster a working, global communications strategy. They manage radio frequencies, standardise international dialing and generally take an interest in all forms of communication. The ITU's approval of H323 has made it the standard for IP-based voice and video communication.

Okay, so that's a nice bit of background, but where does OpenH323 come in? Well, being a corporate organisation, the IMTC isn't so fussy about open technologies. Many of the specifications of H323 relied on patented or protected technologies which, while within the power of telecoms companies to licence, rather locked out open source development.

The OpenH323 project was started in 1998, mainly with the support of Equivalence Pty, an Australian company. The aim was to produce an open source stack, compliant with the H323 standard,

H323 standards

The H323 standard is purely concerned with the framework and process of packet-based communications. There are many referenced standards for all sorts of related standards

H323 – Packet-based multimedia communications systems
H.225 – Call control protocol
H.235 – Security
H.245 – Media control protocol
Q.931 – Digital subscriber signalling
H.450.1 – Generic functional protocol for the support of supplementary services in H.323

H323 Annexes

Annex D – Real Time fax over H.323
Annex E – Multiplexed call signalling
Annex F – Simple Endpoint Terminal (SET)

Annex G – Text SET
Annex H – Mobility
Annex I – Operation over low QoS Networks
Annex J – Secure SET
Annex K – HTTP Service Control Transport
Annex L – Stimulus Signalling
Annex M – QSig Tunneling
Annex N – QoS

Audio codecs

G.711 – PCM audio codec 56/64 kbps
G.722 – Audio codec for 7 KHz at 48/56/64 kbps
G.723 – Speech codec for 5.3 and 6.4 kbps
G.728 – Speech codec for 16 kbps
G.729 – Speech codec for 8/13 kbps

Video codecs

H.261 – Video codec for >= 64kbps
H.263 – Video codec for < 64kbps

which would enable small scale developers and open source projects access to this technology.

OpenH323 has addressed the primary structure of the H323 standard, but there are still some issues which are unlikely to be resolved soon. These concern the video and audio 'codecs', the algorithms used to compress and decompress the audio and video data for transmission across the network.

Most of these are based on patented technology and, with the introduction of legislation such as the DMCA in the US, are protected so completely that it's probably illegal to think about how they might work. If you are using H323 hardware (see *Telephony hardware* box) these codecs may be supported, because the manufacturer has licensed them, otherwise, you are going to have to stick to the openly referenced ones.

Fortunately, these are quite good and in fairly common use. Most VOIP

client software will support them, so compatibility shouldn't be too much of an issue. The G.711 and GSM codecs are currently supported for audio and the standard H.261 video codec (although this only really works well under Linux, which has the advantage of the Video for Linux subsystem).

Other open codecs may well be supported. Some seems to exist in a legal grey area, and, where redistribution is an issue, it will not be included in OpenH323. If you want to get hold of the latest version of the library, check out the documentation or help out the project, you can find their website at www.openH323.org

Just to confuse things

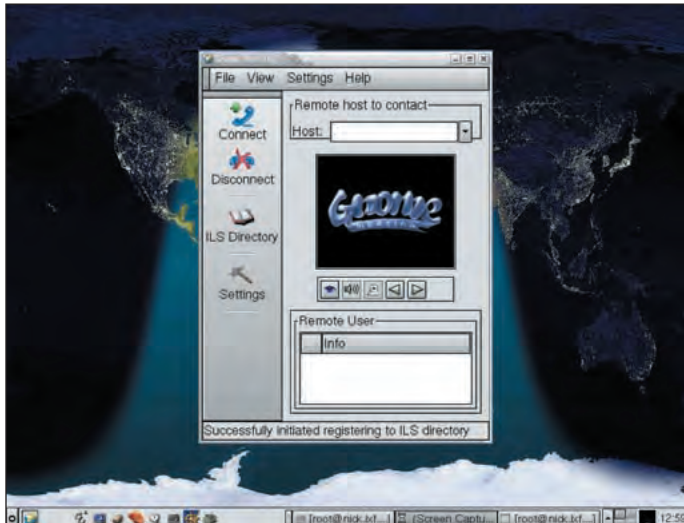
Now that we've learned a lot about H323, you'll be pleased to know that it may be on the way out. H323 is a very far reaching protocol which seems to have thought of everything.

Voxilla

H323 isn't the only standard for communications, and there are plenty of associated technologies which also suffer from a lack of support. The Voxilla website is a useful resource for developers or users of all sorts of telecommunications software. It plays host to a number of important projects such as drivers for communications hardware, codec projects and server applications like hylafax.

The focus of the site is on server oriented projects, but there are also local downloads for conferencing software and some other utilities.





This simple, uncomplicated and pretty small display is your window on a world of communication.

The only trouble is that all of these extra features have resulted in a rather bloaty design. The transmitted packets contain lots of information which is only useful for more advanced features of H323. For simple client-to-client connections, H323 can be horribly inefficient.

So, of course, a new standard has been proposed, SIP. There was much discussion about whether or not SIP would succeed H323, but at the moment efforts seem to be centred around their happy coexistence. Although, that said, many notable applications, including the omnipresent MS *NetMeeting*, intend migrating to SIP in the future.

But it may not come to that. Both protocols have begun addressing their shortcomings and are converging, albeit slowly. At the very least, there should be a high level of interoperability between SIP and H323 systems in the future.

At the moment, there is little hardware that supports even the later revisions of H323, never mind SIP (although we'll be looking at a SIP client later), but if you are considering hardware, it would certainly be worthwhile determining which standards it will support, and what the

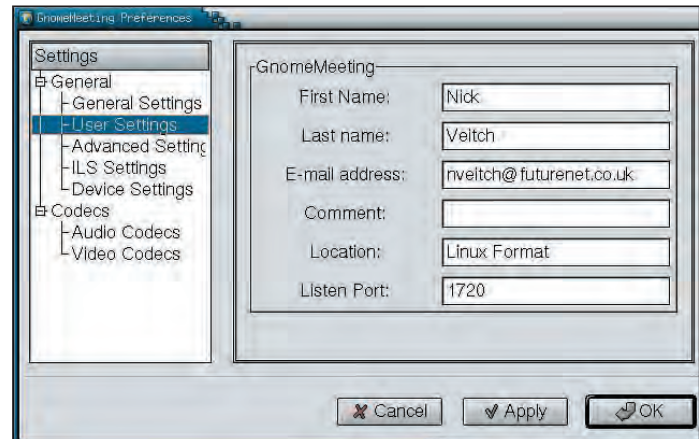
manufacturer's roadmap is for firmware updates.

Gnomemeeting

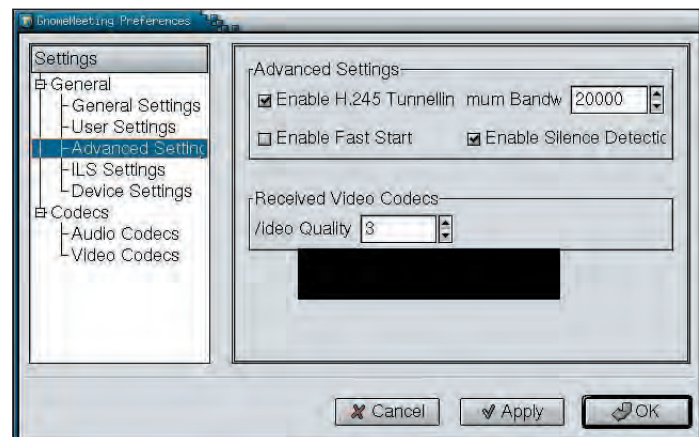
Most certainly the premiere software for video and voice communications on Linux is *Gnomemeeting*. Based around the same ideas as *NetMeeting*, it uses the H323 protocol to transmit and receive voice and video streams over a TCP/IP network.

As it uses the Video4Linux subsystem (see the *Video4Linux* box) and standard audio devices, it supports a wide range of potential hardware you might want to use – if your hardware is supported by Linux, there is a fair chance that it will work with *Gnomemeeting*.

One important hardware requirement though is that your soundcard supports full duplex mode. Sometimes on old cards this is supported by creating two audio devices. On some audio hardware it isn't supported at all, and will be unsuitable for any sort of VOIP software. (You really do need to be able to use the input and output simultaneously for a normal conversation.) Usually, if you have a



Your user settings – needed if you choose to register on an ILS server.



Adjust bandwidth. Enable the H.245 protocol to work through firewalls.

Linux-supported PCI soundcard, you'll be okay – problems tend to lie with 'onboard audio' chipsets and older ISA-based devices. There is a list of devices known to work at the *Gnomemeeting* website.

Getting and installing

The main *Gnomemeeting* website at www.gnomemeeting.org has the latest source and also a variety of packages for popular distros. As well as the *Gnomemeeting* software, you will also need to have the *pwlib* package and *OpenH323*, as well as *LDAP* (used for accessing ILS directories).

If you wish to install from source, there is a useful install guide on the site, and also a very handy *bash* script which will take care of most of the compiling for you. If you haven't already enabled Video 4 Linux in your kernel, this might be a good point at which to do so (though you may want to read about the NAT patch in a few paragraph's time, first).

Video4Linux

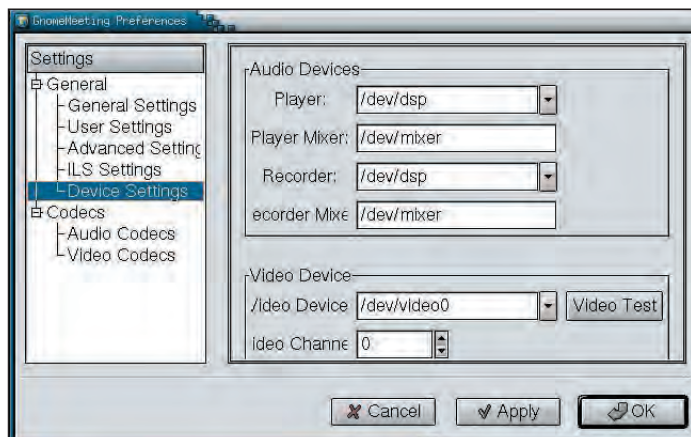
The Video4Linux system supports a number of devices directly or indirectly. Basically, if you want to use a standard webcam device, or any other sort of video capture card in conjunction with a camera, you'll want to enable Video4Linux.

We did a huge feature on Video4Linux back in *LXF20*, so if you are in doubt about how to achieve this, check your back issues. A large number of webcams and standard capture cards are supported (for USB webcams you will also need to enable USB support and include the relevant modules too).

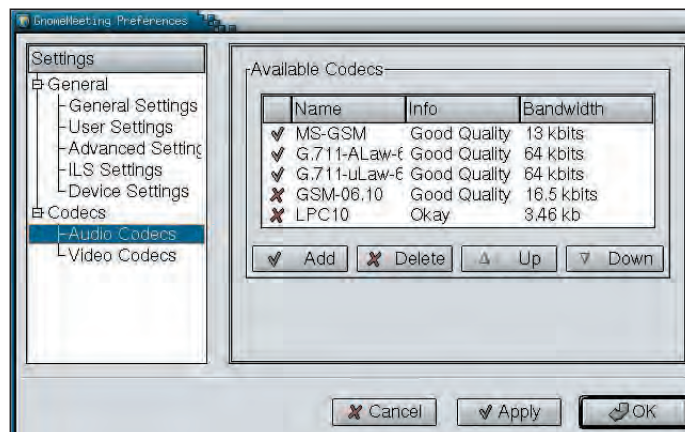
“One important hardware requirement though is that your soundcard support full duplex mode for any sort of VOIP.”



VoiceOverIP



Audio and video devices are usually discovered by *Gnomemeeting*, but if you have several, you may want to adjust them here.



Supported codecs are displayed here - not all are enabled by default, so check which one you will want to use.



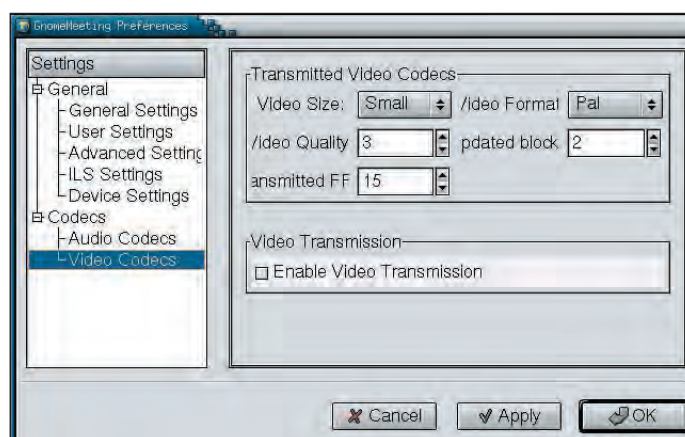
Of course, the software also requires the usual *GNOME* libraries and, of course, it will run under any window system/desktop environment as long as these are present.

Firewalls and other problems

Making a connection when one or other party is behind a firewall poses some special problems. One of these is obviously that the firewall is determined to stop various ports being accessed from outside. The other problem is that firewalls often go hand in hand with NAT and/or proxies – the client at the other end might receive data from you, but all it sees is one IP address for your whole company, plus it runs into a brick wall should it attempt to reply.

One way around this is to use port forwarding. You could connect ports, using *ssh* for example, in the allowable zone to the standard H323 access ports on a static IP machine located outside of the firewall. It would be quite tiresome to do this (have you seen how many ports you need?) and only really effective for one client behind each firewall.

Gnomemeeting supports the H245 tunnelling protocol and will also take advantage of the NAT patch, which should take care of the IP



Video preferences can also be set, but may depend on your hardware. Some webcams will only want to work at certain sizes/framerates.

addressing problem. The patch for 2.4.17 can be obtained from <http://roeder.goe.net/~koepi>

Using ILS

Connecting to other users can be a problem if you don't know where they are! You can use IP addresses, but this is only helpful if the other client has a static IP address and you know they are waiting for your call. *Gnomemeeting* supports the ILS directory service system – log on to an ILS server and you'll get a list of other active clients registered there. Then you can easily pick out the one you want to call by their name or description.

Talking to Windows

It is possible to hook up a call from *Gnomemeeting* to other H323 client software, such as *NetMeeting*. Although, since XP, *NetMeeting* wants to work with "Passport" services, it still has the capacity to use ILS

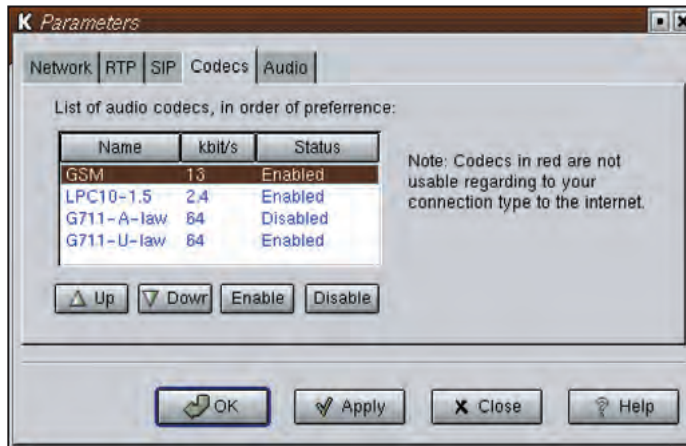
directories or use a direct IP address. One small issue with *NetMeeting* is that since it refuses to list the presence of non-*NetMeeting* clients, the *Gnomemeeting* client will have to call the *NetMeeting* one (when using ILS) rather than the other way around.

Another issue can be the codec support. *NetMeeting* supports a variety of proprietary codecs for audio and video, for which there is no corresponding *Gnomemeeting* support. However, both will support the GSM codec and the standard G.711 codec.

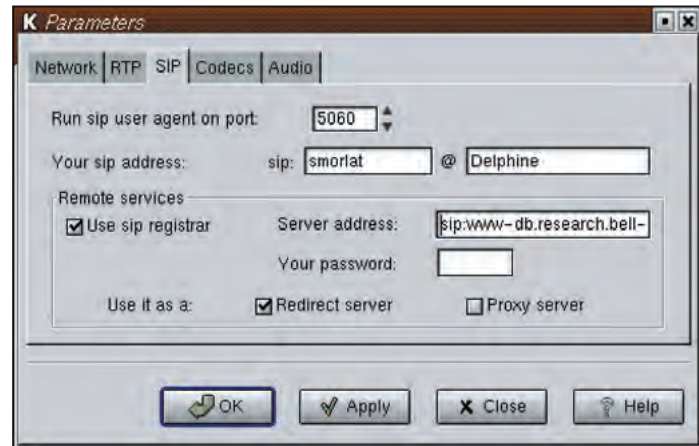
Linphone

If you don't need video and you want to try out the alternative to H323, SIP, then there is a Linux client for you. *Linphone* is designed to be a very straightforward SIP client that supports all the basic features of SIP. As mentioned previously, it seems

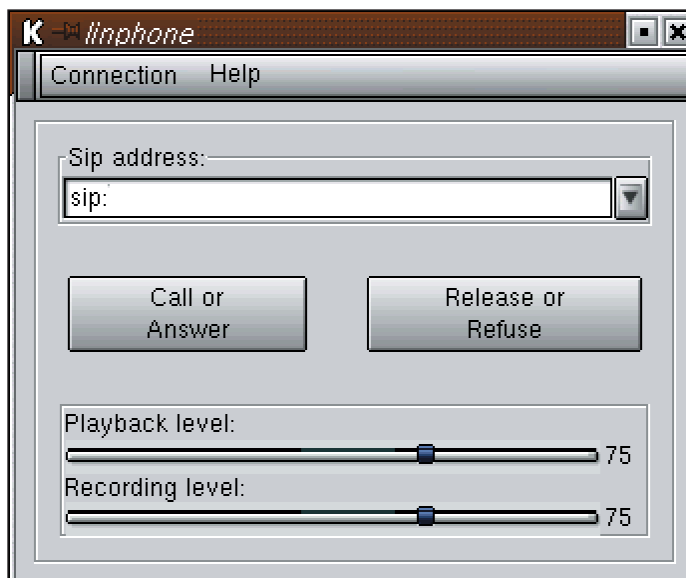
“Making a connection from behind a firewall poses problems. The firewall wants to stop access to various ports.”



Linphone and SIP supports the same codecs as Gnomemeeting over H323.



You can register to a SIP server to allow other users to find you easily.



The simple GNOME interface allows you to make or receive calls.

that future versions of *NetMeeting* will move to the SIP protocol, which may make *Linphone* the only option if you want to talk to your Windows-using colleagues.

You will still need a soundcard of course, and as before the requirement for full duplex is the same. As this is a *GNOME* based application, you'll also need the *GNOME* libraries.

Daemons

Linphone can be run either as an applet (under *GNOME*) or as a silent daemon in *KDE*, to allow it to run unobtrusively all the time and alert you of incoming calls with a phone sound (of course, this is only useful if you have an always on Internet connection of some sort.)

Placing calls is either done by IP numbers as before, or on qualified

domains you can use an address not entirely dissimilar to an email address: sip:green@203.42.118.17 sip:green@green.lxf.co.uk

This looks for the user 'green' at the specified address (the hostname can be resolved to an IP address in the normal way).

Instead of using ILS directories, clients can make use of SIP servers, which serve a similar purpose.

Registering with a SIP server gives you an account there with a permanent server name. So you might become: sip:green@sipserver.co.uk

Each time you run *Linphone* it can contact the SIP server. Calls made to this address are then forwarded to you using your current IP address.

Of course, this method is just as susceptible to the firewall and NAT problems we discussed before in

relation to *Gnomemeeting*.


Linphone can use a SIP proxy (a Linux one is currently in development, though other commercial solutions already exist for other platforms). The proxy will be able to forward calls to users behind the firewall from the outside world (assuming it is set up properly!)

Linphone is fairly basic, but has the features you really need. It supports the same audio codecs as *Gnomemeeting* and has controls for adjusting the volume and mic levels. One slight drawback is that *Linphone* has trouble sharing the audio device, but you can set it to disable the normal soundserver when in use.

Hardware

There is no Linux-supported SIP hardware available at present, but *Linphone* will work with standalone SIP devices. A number of phones supporting the SIP protocol are already in use and can communicate readily with *Linphone*.

Conclusion

It may seem that VOIP is still more or less in its infancy on Linux but, to be fair, the whole technology is in its infancy. With solid clients that support the differing protocols and plenty of support for the open adaptation of the standards, Linux is well-placed for Internet-based telephony. As Linux becomes more and more relevant for servers, it seems likely that the support for specialist hardware and server software will also be forthcoming. In the meantime, it is even now possible to cut your 'phone bills a bit. 



HighAvailability

Highly available networking

John Mehaffey of MontaVista Software explores the implementation of a highly available networking architecture, using the Linux bonding driver and hot swap, to provide for 5 nines of network availability.

High availability means different things to different people. This article will define availability as the percentage of time that a computer system is capable of providing the service that it is assigned to do. A good figure of availability for computer systems that are used for business critical tasks, such as running a telephone switch or enterprise data communication network, is 99.999% of the time (5 nines). This translates to less than 6 minutes per year that the service is not available.

There are many technologies that combine to provide a highly available system, and most of them are based on redundancy, quick failover to spares and quick repair. Compact PCI has traditionally been the platform of choice for these 5 nines systems, although recently clusters of inexpensive PCs have started replacing compact PCI systems in less demanding environments. Hot swap of components into and out of a running system is usually also a requirement.

In a highly available network there should be multiple independent paths to each system in the network to avoid Single Points of Failure (SPOFs). Physical separation is also a good idea since, if both paths are in the same conduit and the conduit gets cut by accident, the network will go down.

In **figure 1**, the hub is a SPOF since, if the hub fails, all paths between System 1 and System 2 will be unavailable. In **figure 2**, there are no networking SPOFs, since any single networking component can fail and there will still be a path available between System 1 and System 2.

The Bonding Driver

Key to availability in **figures 1** and **2** is the ability to quickly detect failure and transparently switch from one LAN connection to another. Putting the burden of handling redundancy in the networking driver allows for easier HA hardening of networked applications, as it relieves the application of having to be aware of network topology.

The Linux bonding driver

(www.sourceforge.com/projects/bonding) has the ability to detect link failure and reroute network traffic around a failed link in a manner transparent to the application. It also has the ability (with certain network switches) to aggregate network traffic in all working links to achieve higher throughput. This is sometimes referred to as trunking.

The bonding driver accomplishes this by enslaving all of the ethernet ports in the bond to the same MAC (hardware level) ethernet address, which ensures the proper routing of packets across the links. With a hub arrangement, there should not be more than one link with the same MAC address active at any one time, so the bonding driver can be set up to have only one channel active at a time. This is called the active-backup mode, and it will route all traffic through that channel until it detects a failure, at which point it switches to the next backup channel in a round-robin fashion. In this way, the higher levels of the networking stack do not see the failover, and things such as sockets just continue to work correctly, even though the hardware they were connected through has failed.

With a switch instead of a hub, it is possible to send traffic over all live links at the same time, effectively aggregating the bandwidth of the available links. This is called the "round-robin" mode. Round-robin mode provides availability as well as aggregation, but not all switches are capable of supporting aggregation. The bonding documentation contains a list of some switches that do support aggregation. The round-robin mode sends packets over all working links, with each successive packet being sent over the next link in the bonding rotation, effectively aggregating the bandwidth of all usable links.

In addition to the mode, the bonding driver has a configuration parameter called MIIMON, the interval for checking link status. The bonding driver checks the status of the link every MIIMON milliseconds, and if the link status is not UP, rotates the active link to the next available slave in the rotation. The value used for MIIMON is dependent on the application, but a good starting value is 100 milliseconds.

The program that creates the bond is the *ifenslave* program. It is similar in function to the *ifconfig* program that configures non-bonded ethernet interfaces, except that it configures all members of the bond to the same network configuration (IP, MAC, broadcast addresses, etc.). To configure the bonding driver, use *ifconfig* to configure the bond0 device and use *ifenslave* to configure the members of the bond (the slaves).

Many recent distributions, including the Hard Hat Linux HA Framework 2.0 release, come with bonding and *ifenslave* already in the distribution. It is also possible to get the bonding driver from the SourceForge web site mentioned above. Bonding is available as a patch that contains the bonding driver and the *ifenslave* program, as well as some other modifications necessary to make the whole package work properly. The bonding driver can be installed into the system either compiled into the kernel or as a module using the *insmod* program.

The following is a typical configuration scenario.

```
sys1# insmod bonding.o mode=1
miimon=100
sys1# /sbin/ifconfig bond0 10.0.0.1
up
sys1# /sbin/ifenslave bond0 eth0
sys1# /sbin/ifenslave bond0 eth1
sys1# /sbin/ifconfig
```

**“A good figure of availability is 99.999%
This translates as less than 6 minutes per
year that the system is unavailable.”**

```
bond0  Link encap:Ethernet
HWaddr 00:A0:CC:D0:DA:10
        inet
addr:10.0.0.1 Bcast:10.0.255.255
Mask:255.255.0.0
        UP
BROADCAST RUNNING MASTER
MULTICAST MTU:1500 Metric:1
        RX
packets:7224794 errors:0
dropped:0 overruns:0 frame:0
        TX
packets:3286647 errors:0
dropped:0 overruns:0 carrier:0
        collisions:0
txqueuelen:0
```



HighAvailability

eth0	Link	inet
encap:Ethernet HWaddr		addr:10.0.0.1 Bcast:10.0.255.255
00:A0:CC:D0:DA:10		Mask:255.255.0.0
	inet	UP
addr:10.00.0.1 Bcast:10.0.255.255		BROADCAST RUNNING SLAVE
Mask:255.255.0.0		NOARP MULTICAST MTU:1500
	UP	Metric:1
BROADCAST RUNNING SLAVE		RX
MULTICAST MTU:1500 Metric:1		packets:3651769 errors:0
	RX	dropped:0 overruns:0 frame:0
packets:3573025 errors:0		TX
dropped:0 overruns:0 frame:0		packets:1643480 errors:0
	TX	dropped:0 overruns:0 carrier:0
packets:1643167 errors:0		collisions:0
dropped:0 overruns:0 carrier:0		txqueuelen:100
		Interrupt:9
		Base address:0x1400

work in the bonding system, but most modern ethernet devices have this capability. Note that all devices in the bond have the same IP and MAC addresses.

The networking stack talks to the bond0 device, which sends packets out over whichever slave device is appropriate given the mode and availability status. In the example above, the mode is active/backup and the active ethernet device is eth0. Inactive ethernet slaves have NOARP in the status line.

Hot Swap

When a component fails, it is not enough to detect and mask the failure, the failing component must be repaired so that the next failure does not cause loss of service. For an ethernet cable or hub or switch, it is usually simple matter of replacing it with a working one. For an ethernet board in a running computer, it is not always so simple.

To repair hardware in most computers, it is necessary to turn off the affected computer, replace the defective board, and reboot the system. During the time it takes to diagnose and repair the problem, there is a loss of redundancy, and unless the service being provided is capable of migration to another system, the repair process will cause a loss of availability as well.

The PCI Industrial Computer Manufacturers Group (PICMG) has created a set of standards for Compact PCI hardware and software that make it easier to replace defective hardware in a running system. With PICMG compliant hardware and the proper drivers and daemons, replacing a defective board in a running system is a simple matter of removing the defective board and replacing it with a working one.

PICMG standard 2.1 is a hardware standard that covers the mechanical and electrical requirements necessary to remove and/or plug in a board in a running system (hot swap). PICMG standard 2.12 is a software standard that covers the driver requirements to handle hot swap events.

The SourceForge PICMG hot swap site (www.sourceforge.com/projects/picmg212-hs) has the hot swap driver routines and *HA daemon* for handling hot swapping.

“During the time it takes to diagnose and repair the problem there is a loss of redundancy and availability.”

collisions:0
txqueuelen:100
Interrupt:10
Base address:0x1080
eth1
Link
encap:Ethernet HWaddr
00:A0:CC:D0:DA:10

The first *ifconfig* sets the IP address for the bonding driver. The next two *ifenslave* commands enslave eth0 and eth1 to the bond0 device.

The bond0 device takes the MAC address of the first slave configured in the bond and this becomes the MAC address for all devices in the bond. Ethernet devices must be capable of changing their MAC addresses to

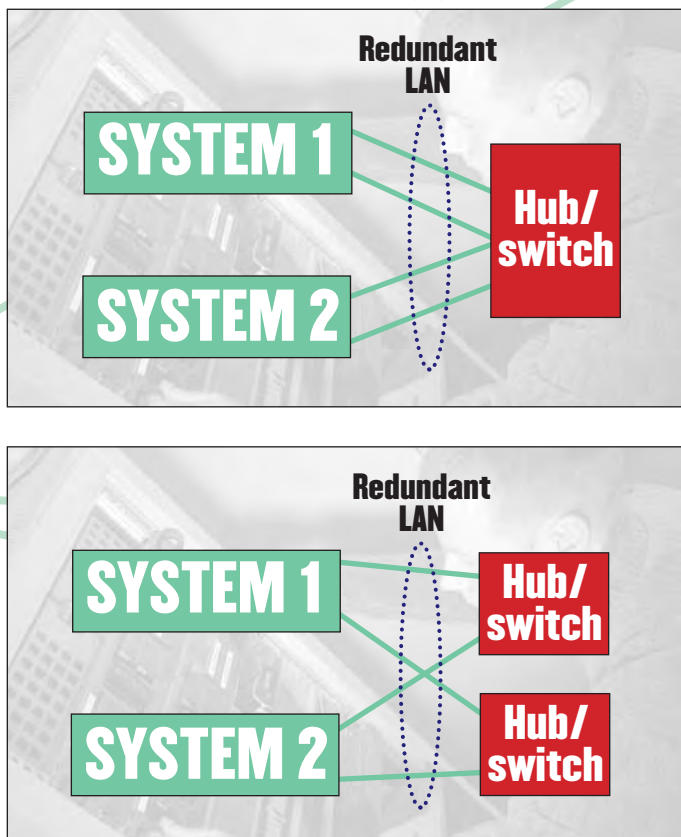


Figure 1 (above right): The hub is a single point of failure (SPOF).
Figure 2 (right): Any networking component can fail – there will still be a path available.

Hot swap requires additional coordination with drivers and the PCI subsystem to handle PCI devices that come and go. When an ethernet card fails and the operator wants to remove it, all he has to do is open the handle switch on the Compact PCI board, and this sends an **ENUM#** interrupt to the PICMG 2.12 driver, which calls to the routine registered to receive hot swap events. This routine is responsible for notifying the driver for the card, removing the device from the kernel PCI tree, and turning on the blue hot swap LED on the board, which indicates to the operator that it is safe to remove the card. It also notifies the *HA daemon* so that it can do any user space actions necessary (such as removing an ethernet device from a bond, or removing a driver that is no longer used).

When a replacement card (or just an additional card) is inserted, it also causes an **ENUM#** interrupt, which gets routed to the same routine mentioned above. This routine is then responsible for inserting the device in the kernel PCI tree and notifying the *HA daemon* that a new device has been inserted.

HA Daemon

The *HA Daemon (HAD)* is a user space program that receives events from the hot swap subsystem. It takes two configuration files, one to specify which devices are supported (and their corresponding drivers), and one to specify actions to take when a hot swap event is received.

If the hot swap subsystem receives an insert event and does not have a driver loaded for the card that was inserted, it sends a "load driver" message to the *HAD*. The *HAD* checks its device driver configuration file (*/etc/pcidrvrs.conf*), and if it knows the driver for the card, it loads it. If the card is unknown, the *HAD* just ignores the insert event.

The *HAD* also has another major duty with regard to hot swap, and that is configuring the card that has just been inserted. For example, if the card is involved in networking, it needs to have its address established, or if it is a member of a bond, it needs to be enslaved.

The *HAD* accomplishes this through the use of another configuration file, the *HAD*

configuration file (*/etc/had.conf*). This file has a set of configurations and slot/device combinations that use these configurations. *had.conf* is set up by the system integrator or system administrator.

Sample */etc/had.conf* file:

```
# Section 1 - domain ownership
#
domain a on
domain b on
#
# Section 2 - configurations
#
bond bond0 10.0.1.1
eth config1 manual 10.0.1.1
255.255.255.0 10.0.1.0 10.0.1.255
eth config2 bond bond0
#
# Section 3 - slot assignments
#
device 2 0 eth config2
device 2 1 eth config2
device 12 0 eth config2
device 12 1 eth config2
device 16 0 eth config1
```

This file is for a Motorola 8216 chassis with two I/O domains. The first two lines in section 1 state that this processor is going to control both I/O domains. A chassis with only one I/O domain may skip this section.

The first line in section 2 indicates that the *HAD* will start the bonding driver for bond0, and configure it with IP address 10.0.1.1.

The next two lines define ethernet configurations that will be used by the ports in the boards described by section 3.

Configuration config1 is an example of a non-bonded ethernet configuration. It has four parameters, the IP address, network mask, network address and broadcast address.

Configuration config2 is an example of a bonding configuration that will enslave any board that uses it to the bond0 device configured by the bond command in section 2.

The remainder of the *had.conf* file states which configurations are used by devices in the backplane. The first parameter of the device command is the slot and the second is the sub-device. Thus, the card in slot 2 is a dual ethernet card and both ethernet ports will be enslaved to bond0. The device in slot 12 is also a dual ethernet card and the device in slot 16 is a single ethernet that will be configured with the



non-bonded configuration specified by config1.

Cards that are inserted in a slot will not be configured by the *HAD*, but will have their driver loaded if it is in the */etc/pcidrvrs.conf* file.


Summary

The Linux bonding driver can be an important component of a highly available system and, coupled with the hot swap capability of Compact PCI hardware, is capable of providing networking with 5 nines of availability.

Future Directions

The bonding driver could use a number of improvements. It only detects link failure through the ethernet link status indicator and could use a mechanism to diagnose more subtle failures so that availability is not impaired. Perhaps a monitor that periodically sends traffic out the link to check for connectivity could send a link failure message to the driver through some sort of management API. The bonding driver should also be enhanced to provide monitoring software with an indication of when it has detected a link failure and routed around it so that a repair strategy can be implemented.

The *HAD* could also use some enhancements, for example to configure more kinds of cards and handle more kinds of HA related events.

The beauty of Linux and Open Source is that you don't have to wait for someone else to do it, you can do it yourself! 

Hot swapping is easy – with PICMG compliant hardware and the High Availability Daemon.

About the Author

John Mehaffey is the author of the PICMG 2.12 driver in SourceForge, as well as a participant in a number of PICMG working groups including the 2.12 and 2.13 standards. John works for MontaVista Software as a technical marketing engineer and is also the mayor of Saratoga, California, a city of 30,000 in Silicon Valley. Contact John at mehaf@mvista.com

What on Earth is... DEVFS?

Biagio Lucini explains devfs, the alternative way of managing the entries in /dev offered by the 2.4 kernel series.

>> Once again an obscure term. Tell me, what on Earth is devfs?

Devfs is an alternative way of handling the entries in the /dev directory.

>> What are these entries?

As you know, in Unix and derivatives one of the main ideas is that "everything is a file", included the peripherals and the other hardware components attached to the system. The entries in /dev are just special files, mainly used by the OS to interact with your hardware (or logical devices) in the same way as if every single component were just a file.

>> Wait... how can my mouse be like a text file?

Good point, let's explain with a comparison. You can visualise the content of a file with

cat. So e.g.

```
# cat < myfile.txt
```

will show the content of myfile.txt. Now, as root, try

```
# cat /dev/mouse
```

(replacing, if needed, "/dev/mouse" with the actual device corresponding to your mouse, e.g. /dev/psaux) and start to move the mouse. You'll see displayed symbols needed for your OS to convert your movements into a change of the position of the cursor (type **C** while holding the CTRL key when you want your mouse to respond as usual).

>> This is a cool feature. But is it of any practical use?

Not only are these special files useful, they are fundamental — it is by means of special operations

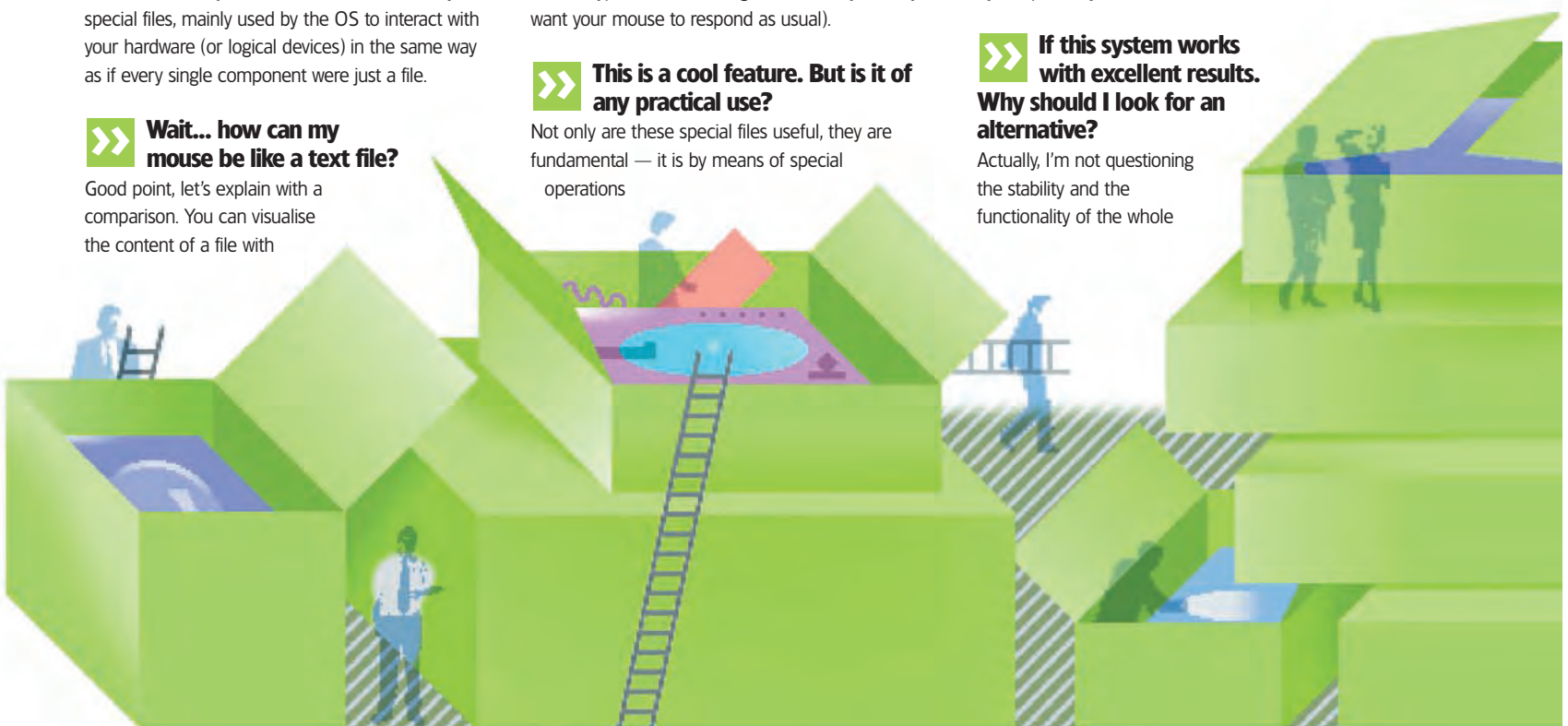
performed on them that low level modules access your hardware. Such an approach is elegant and practical at the same time. In principle, it treats in the same way files physically residing on the file system and physical devices attached to the system, so that you can use the same system calls when accessing a text file and when accessing a scanner.

>> How can my hard disk be associated to a file?

The way of associating physical or logical devices to a file is through the creation of a special entry in the directory /dev. These devices are essentially of two types: block (buffered) and character (unbuffered). Each entry is characterised a major and a minor number. The major number univoquely identifies the type of physical or logical device attached. The minor number is a label introduced to make more flexible the association between a kernel driver and a device, and its meaning depends on the actual device it refers to. For instance, the master IDE hard disk connected to the primary bus corresponds to a block device with major number 3 and minor number 0 located in /dev and named hda, the first partition on it corresponds to the block device with major number 3 and minor number 1 (hda1) and so on. The minor number 64 is whole slave disk, with partitions identified by minor numbers from 65 to 127. In other cases the minor number identifies the physical/logical device of a given type, e.g. UART serial ports corresponding to the character devices /dev/ttyS[0-191], have major number 4 and minor number ranging from 64 to 255. When a special file in /dev is accessed, the driver to be used is identified by the pair major/minor number.

>> If this system works with excellent results. Why should I look for an alternative?

Actually, I'm not questioning the stability and the functionality of the whole



implementation. The alternative that I'm discussing works according to the same fundamental principles, but offers in addition a solution to present annoyances and foreseeable problems. Look for instance at your `/dev` directory. How many entries does it have? How many of them do you need for your system? This translates in a waste of space, and although today hard disks are cheap and you as a desktop user might not care, this issue can be fundamental for embedded systems. In addition, every time a new piece of hardware hits the market, a new entry in `/dev` may be needed. Leaving aside for a moment the problem of creating a device (via the `mknod` command if you know exactly how to do that or the more friendly `/dev/MAKEDEV` script, which may not help in all cases), a new device requires the allocation of major and minor numbers, which are arbitrary as long as they don't conflict with any other existing device. A first consequence of this is that a database of registered device must be maintained (the file `Documentation/devices.txt` of the kernel source tree contains a list of all Linux allocated devices) and major and minor numbers must be officially issued. Major and minor numbers are presently limited to 8 bits — this will come to be a severe limitation. While in principle it is possible to increase the size of these numbers, I gather that in practice it is not straightforward to modify the corresponding kernel code in a clean way.

>> I see, there is no problem for the present, but there will be in the future. So I can stick with the standard implementation until the events will force me to change?

That's right, you can, *devfs* is absolutely optional for the moment. But it is also a clean implementation of the "everything is a file" philosophy that can be worth a go before it forcibly enters our "every day" life.

>> So far you've spend some time praising devfs, but you haven't told me yet what it is.

To quote the *devfs* developer Richard Gooch, "*devfs* is an alternative to 'real' character and block special devices on your root file system. Kernel device drivers can register devices by name rather than by major and minor numbers. These devices will appear in *devfs* automatically..."

Traditionally a kernel device driver registers a device with the rest of the system through calls to the functions `register_chardev()` or `register_blkdev()`. With *devfs*, this two functions are replaced by `devfs_register()`, which accepts as argument the full path of the device as it should appear under the `/dev` directory. This gives flexibility and allows an hierarchically cleaner structure of `/dev`. The required node is created under `/dev` in the sub-location requested. While it is also possible to pass to `devfs_register()` major and minor number for compatibility with the standard approach, this is no longer needed — now devices are associated to the drivers that registered them by name, rather than by major and minor number. As soon as a device is not needed, a call to `devfs_unregister()` removes the corresponding entry from `/dev`.

>> Hang on, back-up a minute. If I decide to go back to the traditional approach, what do I find in `/dev` directory?

Relax, from this point of view *devfs* is very safe. *Devfs* is a

file system mounted on `/dev` (or indeed anywhere you like, though `/dev` is where many programs expect to find it). As soon as you choose not to mount it, you get your old `/dev` directory back in its place, without modification at all.

>> It sounds really clever. But are the only advantages aesthetics and the optimisation of the space?

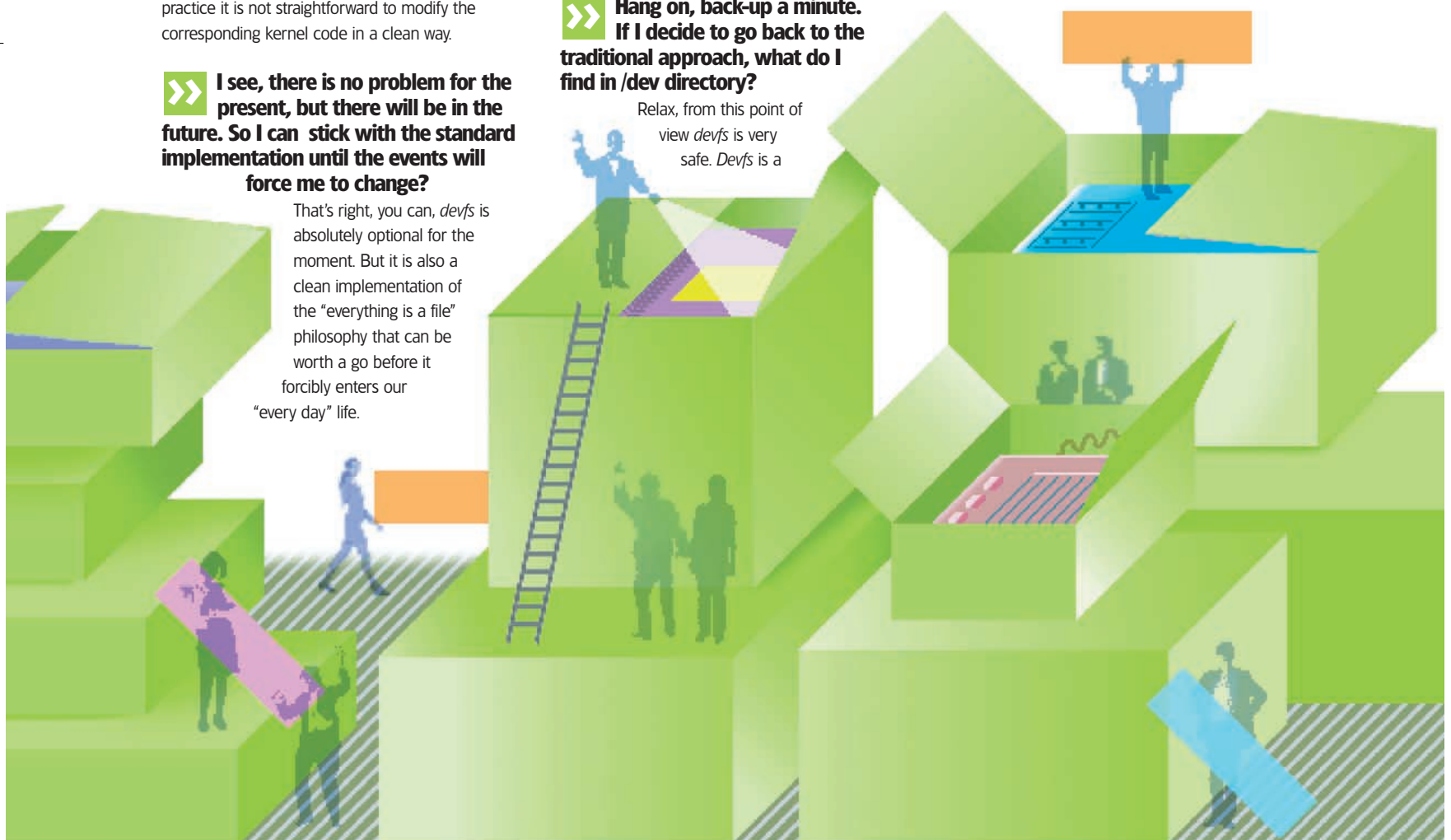
Not at all. First of all, it can easily implement a more logical structure of the `/dev` directory on the fly, via calls to `devfs_register()`. Then, some time can be saved while running speculative programs that scan `/dev` for guessing the attached hardware. Last but not least, *devfs* provides a faster connection between a Virtual File System node and the corresponding device driver.

>> And all this at what cost?

Almost none. The increase in the size of the kernel or in memory usage is negligible.

>> What are the system requirement to enable devfs?

Though still marked as experimental, *devfs* is in the kernel code since version 2.3.46. Any 2.4 kernel



WhatOnEarthDevfs

◀◀ should be OK, though usually the latest version ensures that possible bugs have been fixed. As regards other packages, make sure you have at least *glibc 2.1.3*, *util-linux 2.10.h* and *XFree86 4.x*. All modern distributions have those prerequisites.

◀◀ Then I have to recompile the kernel, I suppose.

That's right. The relevant options are `CONFIG_EXPERIMENTAL=Y`, `CONFIG_DEVFS_FS=Y` and `CONFIG_DEVFS_MOUNT=Y`. It is also recommended to disable the support for the `/dev/pts` file system for Unix 98 PTYs, since it may not work correctly and anyway *devfs* offers similar functionalities. If you remove this support, don't forget to comment out the corresponding entry in `/etc/fstab` before rebooting. Then recompile and install your kernel and modules. Don't forget the modules, otherwise things won't work as expected.

◀◀ And now can I just reboot and enjoy the new devfs?

Careful, for once the kernel is not the end of the story (actually, in this case, recompiling the kernel is the easiest step). Of course a working kernel is still a prerequisite, so make sure that it can boot correctly with the old style

`/dev`, by passing to your boot loader the option `devfs=nomount`.

◀◀ What more is required?

Essentially some tweaking of the initialisation scripts (which requires a more than elementary knowledge of *bash* programming) and the configuration of a companion daemon, *devfsd*, which takes care of compatibility with the old name scheme.

◀◀ It's starting to be tricky...

It may seem so, and actually there is some possibility that you end up with an unbootable system if something goes wrong. As usual, it is better to be prepared to this possibility before it is too late. Firstly, I would recommend you experiment on a test system. Then learn how to recover a broken system (if you don't know that). Finally, remember that it is wise to backup and log every modification to the initialisation scripts, so that if needed the original state can be restored. But don't be put off before you actually know what to do to get *devfs* working!

◀◀ Well I bravely accept the challenge. Are there other issues I should know before starting?

Just one. It seems that the PAM authentication method does not work correctly. To overcome the problem, you should append the lines

```
vc/1
```

```
vc/2
```

```
vc/3
```

```
vc/4
```

```
vc/5
```

```
vc/6
```

```
vc/7
```

to your

`/etc/security` file. Don't worry about security: with this addition, your box is as secure as it was before.

◀◀ Let's start the big boy job, then! What shall I do?

First, get *devfsd* from <http://www.atn.csiro.au/~rgooch/linux>, compile and install it. Note that it comes with a default configuration file. The second step is to hack a little the system initialisation script. What is in this script in your case depends on the distribution you have, particularly on the standard it conforms to. On SysV systems, it's `/etc/rc.d/rc.system`, on BSD-style systems it is usually `/etc/rc`. At the very beginning of this script, write the line

```
/sbin/devfsd /dev
```

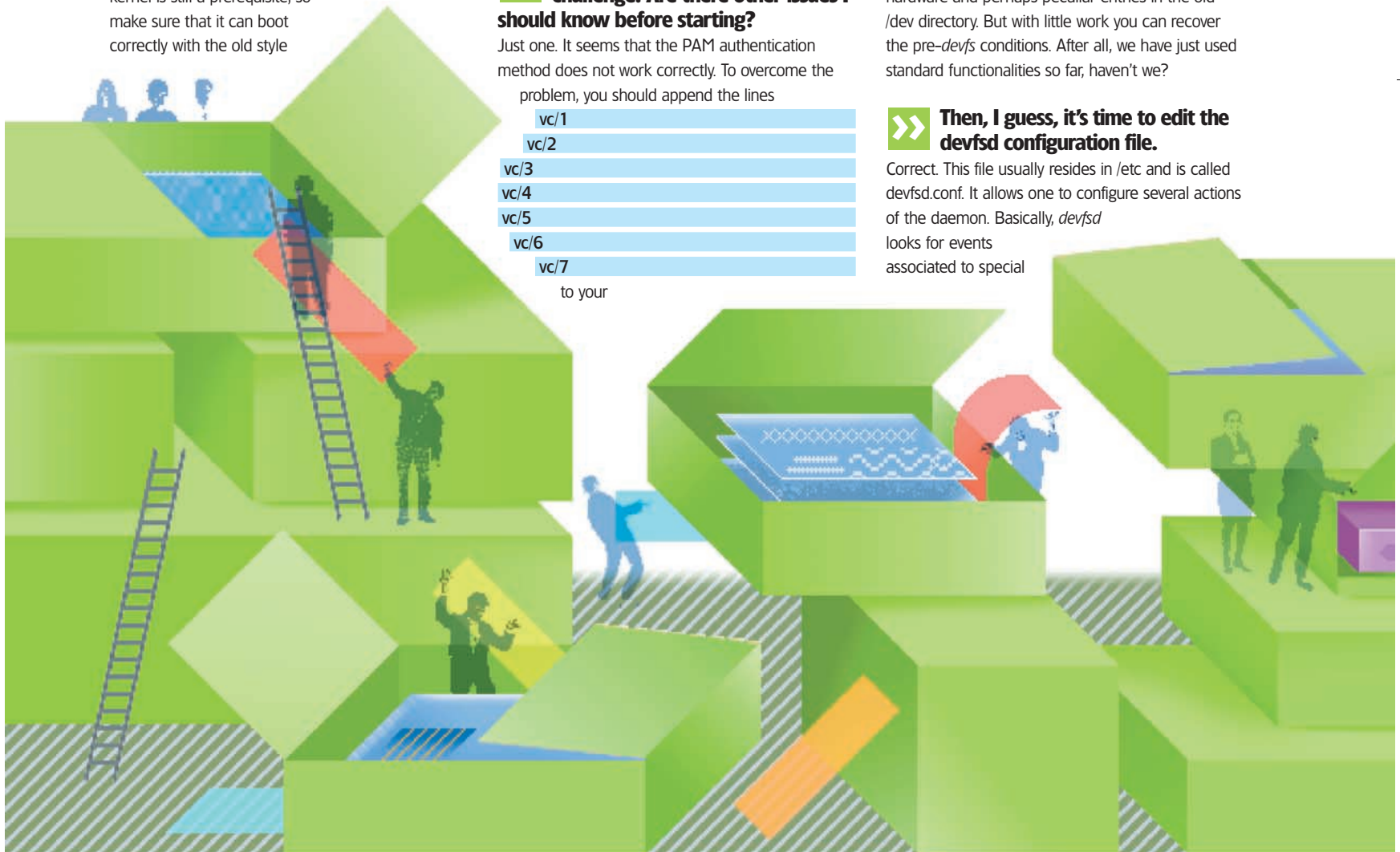
It is very important that this script is run from the very beginning, since it provides entries in `/dev` for the old-style devices which may be needed by some process called by the system initialisation script.

◀◀ Is that all what I need to do? It could have been worse...

It's not that bad, but we haven't finished yet. After this modification, don't expect your system to work perfectly, because every system has its own hardware and perhaps peculiar entries in the old `/dev` directory. But with little work you can recover the pre-*devfs* conditions. After all, we have just used standard functionalities so far, haven't we?

◀◀ Then, I guess, it's time to edit the devfsd configuration file.

Correct. This file usually resides in `/etc` and is called `devfsd.conf`. It allows one to configure several actions of the daemon. Basically, *devfsd* looks for events associated to special



devices (recorded by *devfs* in the file */dev/devfsd*) and every time the event matches some entry in its config file it will take the corresponding action.

» What is the structure of an entry in *devfsd.conf*?

Every entry is structured as follows:

```
EVENT DEVNAME ACTION par1 par2
```

where EVENT identifies the particular event (acceptable entries are REGISTER, UNREGISTER, LOOKUP, CLOSE, CHANGE, CREATE, DELETE), DEVNAME is the name of the device to which the event is associated (it can be also a regular expression) and ACTION is the action to be taken, which can require some input parameters.

» Can you give me an example?

Suppose that your USB scanner in the old name scheme is accessed under */dev/usbscanner*. In the new it is */dev/usb/scanner*. Some application may still expect to find a device */dev/usbscanner*, so you may need a symbolic link. *Devfsd* can handle the creation of symbolic links. The appropriate line is

```
REGISTER usb/scanner CFUNCTION
GLOBAL symlink $devname usbscanner
UNREGISTER usb/scanner CFUNCTION
GLOBAL unlink usbscanner
```

Practically, we are telling *devfsd* that every time a driver registers the */dev/usb/scanner* device we want a symbolic link */dev/usbscanner* pointing to that device. The modifications to */etc/devfsd.conf* will take effect after the daemon has been restarted with

```
# killall -HUP devfsd
```

The above technique can be used if your distribution expects links like */dev/mouse*, */dev/modem*, */dev/cdrom* etc.

» What else can I do with *devfsd*?

Devfsd is powerful and versatile. It allows to create old style names, new style names and symbolic

links, changing the standard permissions, loading the required modules (via the file */etc/modules.devfs*) & executing built-in functions and even customised scripts. It is also possible to use regular expressions to indicate devices — e.g. an entry in the second column like *vc/** means the corresponding action must be taken for all devices in */dev/vc*. As a trivial example, the ability to execute scripts can be exploited to have icons on the desktop appearing and disappearing respectively when you plug and unplug some USB peripheral.

» Have all the device drivers been ported to the new *devfs* scheme?

No, but the most fundamental ones have. A list of all ported devices is in the Readme in Documentation/filesystems/devfs/ off the kernel source tree.

» OK, then, suppose that I need some device drivers that have not been ported yet. What shall I do?

You should create the appropriate device name with *mknod* (I guess that *MAKEDEV* won't help in this case).

» Provided that I know how to do that, how can I make changes permanent?

A quick and dirty way could be to insert the appropriate *mknod* command on the initialisation script. There is a more elegant way, that again passes through *devfsd*. Add to the configuration files the following lines:

```
CREATE * COPY $devpath
/dev-state/$devname
DELETE * CFUNCTION GLOBAL unlink
/dev-state/$devname
```

This will copy all newly created devices to the directory */dev-state* (which should exist) and remove them from the same directory when they're deleted. Now, at the very beginning of your modified system initialisation script add the line

```
cp -af /dev-state/* /dev > /dev/null 2>&1
```

so that at every reboot all special devices you

need will be immediately available in */dev*. If you don't know how to create the devices, but in your old system you have them available, then just copy them in */dev-state* while you have access to the old */dev* directory, and by virtue of the above line they'll be available also when you use *devfs*.

» And if I want my old */dev* directory available anyway?

You can still mount it, but this must be done before *devfs* is mounted. First of all, boot with the option **devfs=nomount** (add the line

```
append=" devfs=nomount"
```

to your */etc/lilo.conf* in order to make it permanent and then run */sbin/lilo -v*). Then, provided that you want to mount the old */dev* in */dev-state*, your system initialisation script must start with the lines

```
mount --bind /dev /dev-state
```


```
mount -t devfs none /dev
```

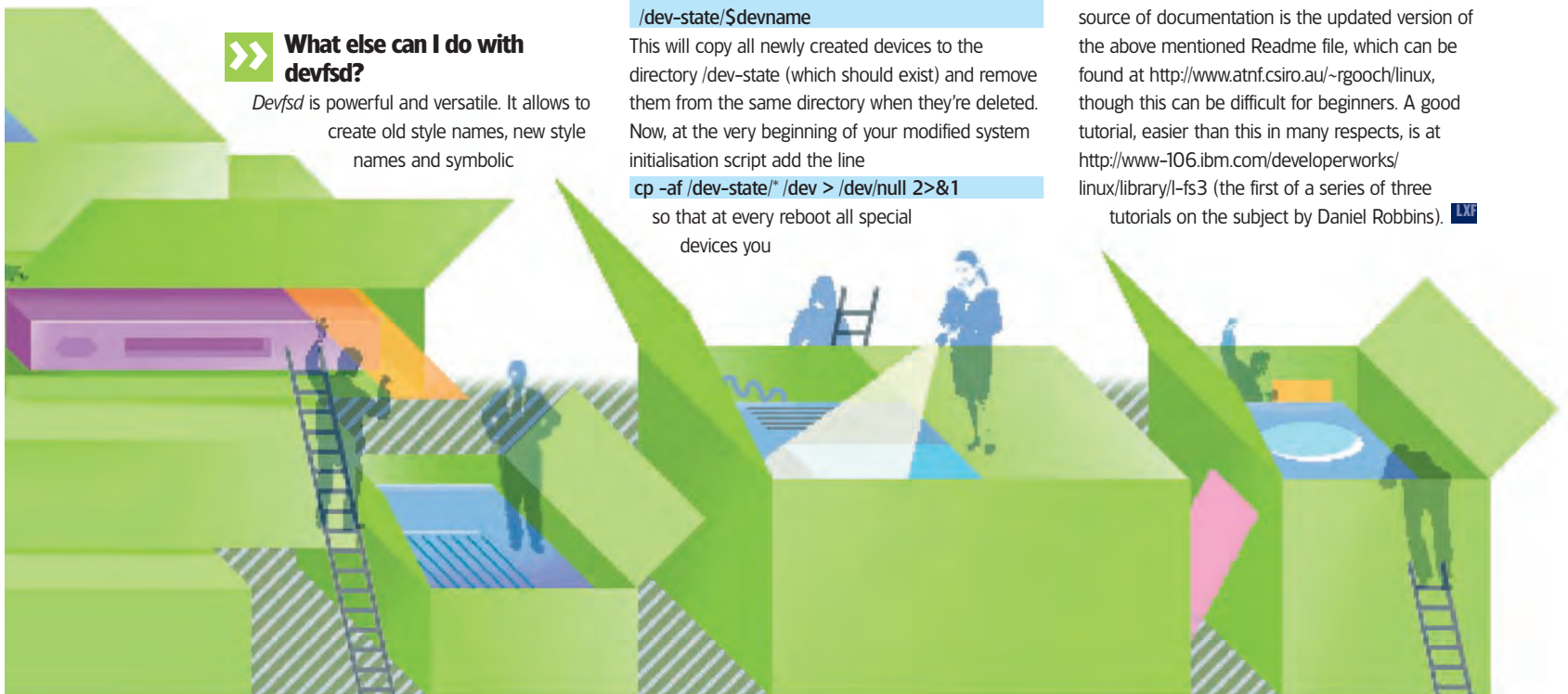
```
devfsd /dev
```

» Uhm, perhaps there is an easy way of getting things working: namely, that somebody has already done it for you. Are there any distributions that offers *devfs* by default?

Among the most popular distribution, the only one with *devfs* support I'm aware of is Mandrake 8.1. Another less-known distribution which offers *devfs* support is Gentoo (<http://www.gentoo.org>), which was on our coverdisc some time ago.

» Are there good tutorials on the net on the subject?

Not that many at the moment, I'm afraid. A good source of documentation is the updated version of the above mentioned Readme file, which can be found at <http://www.atnf.csiro.au/~rgooch/linux>, though this can be difficult for beginners. A good tutorial, easier than this in many respects, is at <http://www-106.ibm.com/developerworks/linux/library/l-fs3> (the first of a series of three tutorials on the subject by Daniel Robbins). 



Emulators



Acorn emulators

Simon Goodwin tests Linux Acorn emulators from Atom to Archimedes.



Your cover disc includes emulator sources and many of the utilities mentioned in the article.

Acorn's BBC Micro model B became a standard fixture in British schools.

Acorn was a major player in the UK micro industry for 20 years, first selling a bare-board 6502 development system, then home computers of steadily increasing sophistication from the Atom, via the BBC Micro, to the 32 bit Archimedes powered by their own ARM chip, capable of running Linux in its own right.

Few home micro manufacturers develop a microprocessor architecture as well as computers and operating systems. This rare legacy means ARM emulation is still very relevant. A dozen Linux programs allow you to emulate features of all the Acorn desktop systems, from the Atom, through various models of Acorn BBC Micro, to the Archimedes.

Acorn stopped making computers in 1999, but rights to their machines passed to Pace Microelectronics, work continues on their 32 bit operating system RiscOS, and the spin-off company ARM Holdings, formerly Acorn RISC Machines, is a leading processor design company, responsible for millions of chips at the heart of PDAs, Set Top Boxes and other appliances. So Acorn emulation can give an insight into one of the most pervasive RISC architectures, as well as access to much classic British educational software.

Acorn's roots

Acorn-developed BBC Micros were standard in UK schools for more than a decade. These machines, known as Beebs, had a generous range of interfaces, thanks to stipulations of the contract for the BBC's computer literacy series that brought the broadcaster's endorsement.

The token-ring Econet gave effective networking for classroom use — and fully

expanded machines had an MFM floppy controller, RS423 serial, Centronics parallel and four analogue inputs, a user port, 1 M/s parallel expansion bus and a 'tube' port for fast parallel connections to memory shared with coprocessors, through FIFO buffers. Later this was used to augment the 2 MHz 6502 with faster versions, CP/M-compatible Z80s, and other CPUs.

The BBC Micro project was born in 1980 but it took two years and much wrangling before Acorn got the BBC Micro into production, with help from ICL, and there was a six month waiting list for much of 1982. Acorn's previous machines, the Acorn 1 and Atom, were far simpler machines, typically sold as kits. The Acorn 1 was a bare-board 6502 programmed in hex. The Atom added a case, keyboard and flickery TV display, but with only 2KB of RAM, expandable to 12KB, and an 8KB Operating System.

Later Acorn pitched a small version of the BBC Micro, the Electron, at home users, arguing that it would give owners a head start at school. The Electron was a disappointment, slower than the real thing and not entirely compatible, and lacking the games software range of its home computer contemporaries like the Commodore 64 and Spectrum 128.

Commodore responded to the educational dominance of the BBC Micro by bundling a Beeb emulator with their *Class of the Nineties* UK Amiga pack. This made emulation a major selling point of a new computer bundle, and highlights the relevance of the BBC Micro a decade after its introduction. Acorn themselves eased the introduction of the 32 bit Archimedes by producing a competent emulator for old 8 bit BBC micro software.

The Archimedes shrugged off Acorn's original OS and DFS and ADFS filing software in favour of RiscOS, a modern multi-tasking operating system with a neat GUI. A stop-gap, called Arthur, was written mainly in BASIC and looked better than it worked. The best Acorn game is *Elite*, a hybrid of true 3D shooting with space economics in the *Galactic Trader* mould, and *Frak* was the best-looking original platform title. But educational software remains the Beeb's greatest asset.



Eight Acorn emulators

There are eight Linux emulators for eight bit Acorn systems, including one for the Atom, a couple capable of running Electron software, and several for the BBC Micro range, from Model A to Master Compact.

AtomEm

Version 0.3 of *AtomEm* comes as C source, with scant documentation, though the author, Frans Faase, has a helpful web page. *AtomEm* tries to load three 4KB ROM files – ‘akernel.rom’, ‘abasic.rom’ and ‘afloat.rom’, though the last was optional on the original machine. It emulates the Atom’s display, CPU and keyboard. Press shift to reverse the direction of the two supported arrow keys.

A useful monitor and disassembler is built in. HELP lists most of the instructions, but not BREAK – to set a breakpoint – or LOAD and SAVE. *AtomEm* is capable of loading binary programs, HTML or text listings, but not disk images.

AtomEm supports lowres text and 256x192 mono bitmaps (without authentic snow!) on any X display depth. It builds OK on SuSE Linux 7.2 after **make depend** to tailor it for the distro, and adding ‘-L/usr/X11/lib’ to LDFLAGS.

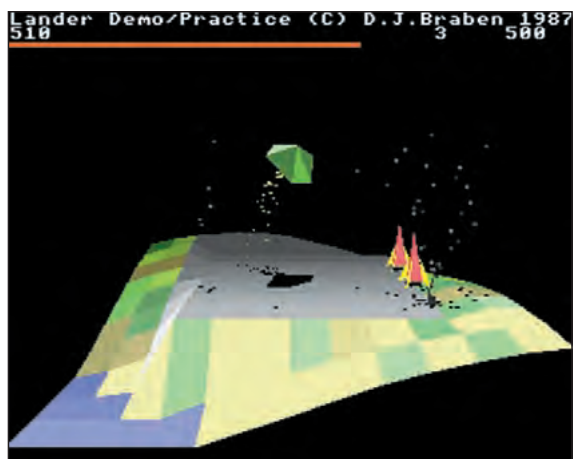
Beeb 6502s

A couple of Beeb emulators focus mainly on the 6502 processor. Version 0.2 of *bbcmicro* comes precompiled for x86 and fails to load *libXm.so.2* when launched. Linking *libXm.so.1* there in ‘/usr/lib’ got it up and running — even without ROMs, though it needs BASIC and OS 1.2 to be useful. It has a good set of monitor commands, and clear resizable displays of the CPU state, memory and 6502 disassembly, but not a lot else.

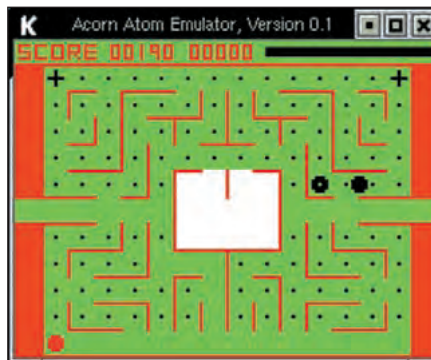
em6502 is a 6502 emulator capable of running BBC BASIC and Acorn’s operating system, written in 1996 as a Bristol University project. To compile and run this on a recent Linux box takes just four commands:

```
cp Makefile.linux Makefile
make depend
make
./em6502
```

This gets the nice text GUI up – but unfortunately text is all you get. *em6502* concentrates on CPU emulation and does not



Lander, later Zarch, was an impressive demo of the Archimedes 32 bit BASIC.



The chunky atom Snapper game is unplayably fast on a modern PC.

simulate other aspects of the BBC micro, like graphics, sound or peripheral devices. Even so it’s capable of running BBC BASIC, in the teletext mode 7.

em6502 needs a patched version of Acorn’s OS ROM; the documentation outlines the patching but doesn’t automate it. Like *bbcmicro*, *em6502* appeals most if you need to write and debug BBC Micro machine code.

beebem

Beebem comes from the author of *Arcem*, and is a front runner despite minor build, display and keyboard foibles. The *beebem* binary comes with a script to load 8 bit X but fails on the *libstdc++-libc6.2-2.so.3* dependency. Source uses **./configure linux**, but cannot find *iostream.h* and moans a lot about it. But **make** compiles to a runnable binary which only falls over for want of ROMs and 8 bit video. It needs 8 bit X with MITSHM.

Graphics are doubled vertically to keep the PAL aspect ratio, and sound is supported, plus disk images in single or double-sided formats with 2.5K per track. Once I added the OS1.2 and BASIC2 ROMs *beebem* worked well with PD disk images I downloaded, accessing them from the default disk path ‘/var/lib/beebem/diskimg/test.ssd’.

Keys are mapped like a real Beeb so ‘’ is next to ENTER and END is the copy key used for two-cursor editing — one picks up characters and the other shows where they will be inserted when END is pressed.

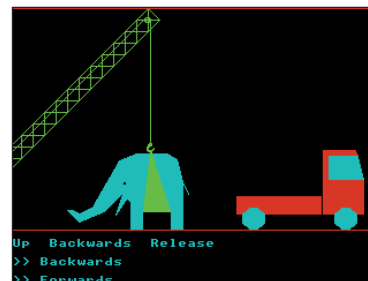
Xbeeb

James Fidell’s *Xbeeb* is sporadically updated but LXF interest prompted a major upgrade earlier this year, when it gained access to most of the formats used for Beeb files online, authentic though CPU-hungry sound, and support for modern graphics depths and resolutions. This long awaited update is on your cover disc, with substantial documentation.

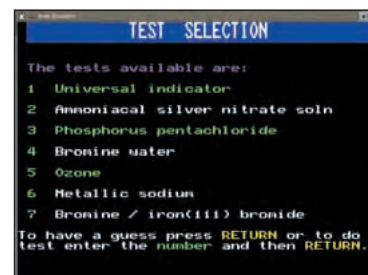
Xbeeb supports access to the Linux file system, with a CATALOG file to hold the information needed by Acorn’s *CAT and *INFO commands. As most of the work is done in Linux not



The 4K BASIC on the Atom was usable but eccentric.



BBC Education — you’ll believe a turtle can control an elephant.



Now you can revise your carbon chemistry with a Beeb.



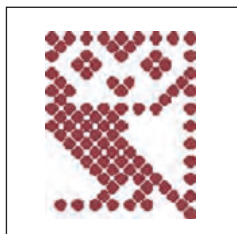
Not SimChancellor, but Great Britain Limited — they don’t make economies like that any more.



Emulators

DFS, filenames are case sensitive; the keypad enter key allows the simulated disk path to be changed. An alternative XDFS using a new emulator specific ROM image is more compatible with a real beeb.

As in *beebem*, the key layout mimics a real BBC micro; F10 is equivalent to F0 on a Beeb, and Insert on a PC keyboard mimics Acorn's copy key. Nine neat X fonts are used for the MODE 7 teletext display, and must be converted from bdf to pcf format before installation. By default Xbeeb runs at the same speed as a real BBC Micro, but the speed limiter is removed if you edit `src/Config.h` to undefine FASTHOST, though this will mess up the audio emulation.



This dotted owl symbolised the BBC Computer literacy project.

beeb

The source for *beeb* has not been touched since 1995 but compiles without problems bar a warning about the old, dangerous and deprecated `gets` function. The emulator needs

'os.bbc' and 'basic.bbc' 15K and 16K ROM files from a real Beeb, saved with something like this:

```
*SAVE OS C000+3C00 C000
FOR I%=84000 TO &7FFF STEP
4:I%:4000=I%:&8000:NEXT
*SAVE BASIC 4000+4000 8000
```

After running these commands in BASIC on a real BBC Micro, put the results with lower-case names and a .bbc suffix in data, and fire up the emulator with `.beeb`. *Beeb* can use cursors rather than X if text is all you want, but support for graphics modes is limited and it has no sound.

TBE

Tom's Beeb Emulator, by Tom Lees, is written in C++. It can emulate BBC Micro Models A and B or Electrons, with ADFS floppy and SASI – early SCSI – drives. It comes with a cross-disassembler for 6502 code and CPU emulation routines. It offers limited sound support, for tones but not random noise.

The version of *TBE* I found lacked serial port emulation and disk formatting code. Tom's TO DO list mentions Electron Plus,

Arc access

Archimedes emulation is a tricky business, given the speed and sophistication of the original machines, but if you're willing to settle for a mid-range ARM 6 with some restrictions it's quite viable on a recent Linux system.

There are four variations on the theme of ARC emulation – *ArcEm*, *ARMphetamine*, *Riscose* and *ROX* – but *ArcEm* is the only one ready for use in classic emulation. The others focus on specific layers of the Archimedes environment, from the ARM processor up to Acorn's RISC OS (Reduced Instruction Set Computer Operating System).

ARMphetamine is a fast emulation project setting out to cross-compile ARM RISC machine code on the fly. Links to its former home on Dynarec appear broken and I've not been able to find anything to try, but if *ARMphetamine* should resurface it could come closer to true ARM performance on alien CPUs than existing interpreting emulators. ARM make their own conventional emulator freely available. It's a good start if you want to run ARM machine code but fairly slow and not specific to the Archimedes hardware.

Linux RiscOS

Riscose takes another tack, aiming to provide a RiscOS environment in which applications can be statically compiled to run on Linux. Developer

Matthew Bloch told LXF, "I'm flat out on other contracts to worry about *Riscose* at the moment, but it has run *grep* for a while, and Norcroft *cc* has been nearly there for a few months. There are a couple of people who chip in occasionally but it's fallen by the wayside for the moment. The implication of running *cc* is that you could cross-compile RISC OS applications on Posix systems."

If you have a lingering enthusiasm for the RiscOS way of doing things try *ROX*. It's not an emulator, but it brings a RiscOS look and feel to Linux. *ROX* is a desktop environment for X which should be instantly familiar to Archimedes users, and quite easy for others to pick up. Once you teach it some file associations it's a capable rival to *GMC* or *KDesktop*, with excellent file matching, searching, localisation and support for drag-and-drop via *XDND*.

ARCEM 0.50

ArcEm is an ARM6 emulator based on code from Acorn, written by David Gilbert, one of the developers of the experimental Amulet ARM processors. *ArcEm* emulates an Acorn A300 or A400 with floppy and hard drives, MEMC memory management mouse and keyboard, authentic video and battery-backed configuration RAM. It has been tested on a 16 and 32 bit X display – earlier

versions required 256 colour palette mapping – but might not work on a packed 24 bit X desktop.

ArcEm 0.50 looks good but lacks sound, Econet, serial and parallel port access. Keyboard handling is rather sluggish and lock keys must be pressed twice to take effect from Linux. Floppies can be in 720KB MSDOS format or Acorn's 800K E format which uses five 1K sectors rather than 9 half K ones, as determined at compile-time. Four files in the launch directory correspond to the floppy drives. These can be links to '/dev/fd0', for example. You must send a signal to indicate a **FloppyImage0** disk change, in order to swap disks without rebooting the emulator.

You need to define hard drives in a 'arcemrc' file in your home directory. The example in the emulator archive allocates a couple of 20MB hardfiles:

```
cp armul-arc/arch/dot.arcemrc ~/.arcemrc
```

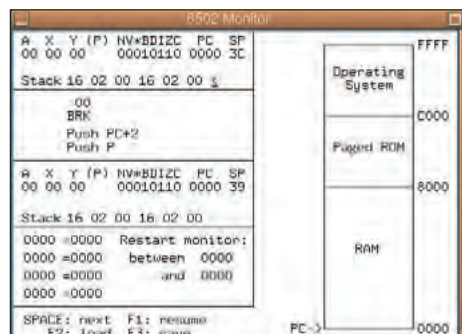
Assign 'ENDIAN=BIGEND' in the Makefile if compiling on a big-endian CPU such as a 68K or PPC. The ARM, like the 6502, the x86 and the author's DEC Alpha, is little-endian.

The makefile looks for the *Gnu C compiler* in '/home/dg/egcs/bin/gcc'; the line 'CC = /usr/bin/gcc' is more likely to work on typical Linux distros. *GCC 2.95* throws up warnings about dodgy definitions and initialisation, mostly in Acorn's ARM emulation code, but it generates a working executable.

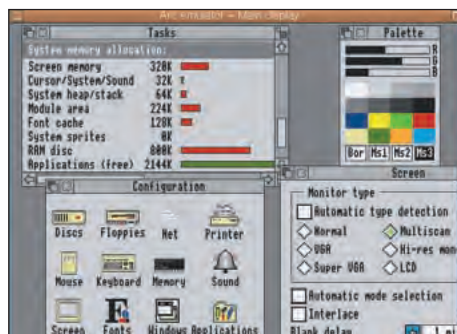
The RiscOS31 ZIP file online contains four 512KB images, and these need to be merged into one 2MB file for the emulator, like this:

```
unzip RiscOS31.ZIP
cat IC24.ROM IC25.ROM IC26.ROM IC27.ROM
>ROM
```

A link to '/dev/fd0' failed to read an Archimedes World disk at the first sector; **CAT** gives **error &41c - disc not understood**. My Amigas can read it with Frank Swift's *XFS* so I guess it's just not in the correct format for the emulator. Acorn switched from D format to E format, which is supported by *ArcEm*, during the nineties, so Arch disk formats vary. MSDOS floppies may be the simplest option.



Em6502 gives a text-mode window into the Beeb's CPU and memory map.



Graphical configuration programs are built into the Archimedes ROM.

Emulators

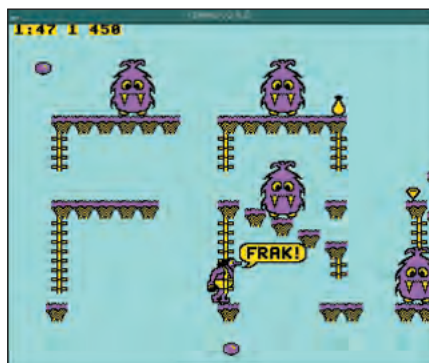
BBC Master 128 and Compact emulation, plus support for tapes, Econet and co-processors — none of which are presently implemented. You'll need *GTK* and thread support to compile the source on Linux.

ElectrEm

ElectrEm is the only Linux emulator specifically for the Acorn Electron. It is available as ready-made binaries with the necessary ROMs, with the knowledge of Pace Micro Electronics who own the rights after the breakup of Acorn.

Unlike the basic Electron emulation in *TBE*, *ElectrEm* supports tape files and +1 and +3 expansion units, and has a menu system. You have a choice of logical or physical keyboard maps, reflecting Electron layout or your host keyboard, or can set your own by editing an ASCII config file. The documentation is in RTF (Rich Text format), with a 19 page generic file and one page about each specific version.

One Linux version uses *GGI* for input and output; the other expects *SVGAlib 1.3*, gaining joystick input but suffering messy ASCII entry for filenames. Sound is a bit erratic on both. Printer



Frak is authentically slow but good-looking in the new XbeeB.



A souped-up Elite now runs natively in Linux via Allegro, and there's even a version that runs within Emacs!

output is text only. If programs are Electron-specific this is the Acorn emulator for you, but if they would also run on a BBC Micro, *XbeeB* or *BeebEm* are more attractive.



The Archimedes has a resident vector graphics package with support for smooth splines.

Once you start the emulator with a RiscOS ROM you will find yourself in a simple shell with a * prompt. The **help** command is a good start; there's a lot of good stuff in that 2MB ROM. *BASIC* puts you into BBC BASIC 5, where the supervisor commands remain available if prefixed with an asterisk. Try ***Help Commands**, and note that, unlike Acorn BASIC keywords, you don't have to type these in capitals. If it freezes, press Scroll Lock once or twice.

The desktop environment is accessed with ***desktop**. Drive icons appear along the bottom, and others as programs are launched. At first you see an Acorn logo, which calls up status information, and a coloured panel to adjust the palette. A folder icon labelled 'Apps' calls up a window with built-in program icons. As you launch these or others from drives they add icons to the toolbar.

You must press + on your numeric pad to lock the Linux pointer in the middle of the emulator window and transfer control to the emulator's pointer, before you can make selections on the Archimedes screen. Press + again to release the pointer for Linux. The middle mouse button calls up RiscOS menus, reminiscent of *KDE2* or *GNOME*'s toolbar, though RiscOS could do it long before.

The main screen background is called the pinboard, and windows appear there with conventional gadgets to move, scroll, pan, raise, lower, maximise, minimise, resize and close these. Pointer tracking is a bit slow but usable; though I found I was not able to make selections on the extreme left of the pinboard, window dragging meant this was not a serious problem.

The ROM includes a vector drawing package, with support for B-splines, a pixel painting application for making sprites and larger images, and the usual desktop calculator, toolbar alarm clock and text editor. The ROM also has context-sensitive help which updates messages in a fixed window depending on what you are pointing at, like Apple's bubble help, but without the delayed floating overlays. The *Characters* application lets you preview the impressive anti-aliased fonts.


Configure calls up a dozen application icons that set up peripherals and GUI preferences. Real Arcs are very good at changing mode on the fly, but *ArcEm* on X has a more difficult job as it must potentially emulate all sorts of display on a single X desktop. I was able to select the colour 800x600 pixel SuperVGA, rather than the 640x480 Multiscan default, and various others by adjusting the monitor

type and flipping though default mode numbers in the screen configuration window. But selecting a HiRes monitor confuses *ArcEm*, which redraws a strip of the screen but then loses track of the mouse.

ROM resources

Your cover disc includes emulator sources, *ROX* Desktop and Filer packages for Debian, rewrites of the classic Acorn space game *Elite* for Linux and, amazingly, *Emacs*, connection and conversion utilities, cross assemblers and disassemblers.

Epic's £5 Flash ROM 2 includes the Atom emulator with software, the Beeb DFS ROM image, and vintage BBC emulators for Unix including three versions of *BeebEm* and *beeb*, *XbeeB* archives and update patches, *em6502* and *bbcim*, a file converter.

It also holds utilities like *Kermit* and other comms software for the beeb, hosted ROM, file and graphic converters, with helpful overview files in the utility directories. Overall Flash ROM 2 contains a relatively well-organised collection of Beeb-related files, though this is far from the sole focus of the CD, which particularly excels in support for another British 6502 system, the Oric. But we'll have to look at those files another month... 

Acorn links

Emulators and ROMs on the web

ArcEm: <http://ftp.linux.org.uk/pub/linux/arcem>

BBC Micro emulator:

<http://www.qualsar.co.uk/bbcmicro>

Beeb emulator:

<ftp://ftp.mrc-lmb.cam.ac.uk/pub/jkb/beeb>

BeebEm home:

<http://titan.abdelion.com/kaell/beebem>

Beeb software: <http://www.nvg.ntnu.no/bbc>

Elite for Emacs:

<http://members.fortunecity.com/salkosuo/elite-for-emacs>

Elite originals:

<http://www.users.waitrose.com/~elitearc/elite/archive>

Electron emulator: <http://electrem.emuunlim.com>

Epic (Flash ROM 2 CD): <http://www.epiconline.co.uk>

Portable Elite in C:

<http://home.clara.net/cjpinder/elite.html>

RiscOS on X: <http://www.janw.easynet.be/rox>

Tom's Beeb emulator:

<http://www.lpsg.demon.co.uk/files/beeb>

XbeeB:

<http://www.cloud9.co.uk/james/BBCMicro/XbeeB>

LINUXformat 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 post, by email (linuxformat@futurenet.co.uk) or log on to our website and post your suggestions in our special forums? (www.linuxformat.co.uk). Hope to hear from you soon!



Nick Veitch
EDITOR

THIS MONTH...

Ext3 >>

We hand out the waterwings for this painless dip in the waters of journalled filesystems **p68**



Partimage

Save your partitions for a rainy day and laugh in the face of disaster with our easy-to-follow tutorial **p70**



<< Python

A practical application for python scripting – a quick and dirty HTML survey generator **p74**

CVS

Find out how to collaborate on large coding projects with the Concurrent Versioning System. We get you started **p78**

Perl

Let your apps talk to each other using inter-process communications with signals, sockets and pipes **p82**

Java

Reflection with Class. Take advantage of Java's self-awareness to write automated development tools and communicate with databases **p86**

Kylix

With this month's CSV browser not only will you learn anchors and constraints – but you will be able to browse our coverdiscs, too **p88**

VisualRoute 5

A useful network tool which doubles as a pretty desktop toy. See our money-saving special offer on upgrading **p94**

SysREQ >>

What does that magic key do? Find out the good and bad of low-level access from the keypad **p95**



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 := not end;
```

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

EXT2 COMPATIBLE FILE SYSTEM

Ext3 Journalled File System



Afraid of migrating to a journaled file system because of the need for a back-up? Then ext3 is the way to go. **Biagio Lucini** shows how simple it is to enjoy the benefit of this journaled file system on your existing partitions.

Among other features, the kernel series 2.4 has introduced support for journaled file systems. The benefit that these bring is that all the changes in the metadata structure of the file system (we recall that metadata are “data about data”, used to describe the file structure on a given file system; it is this metadata structure that actually allows manipulation of files) are logged onto a special file, called a “journal”, and marked as “to be committed” until they have been written to disk. In this way, in the unlucky event of an unclean shutdown the file system recovery process passes through the analysis of the journal and usually takes a few seconds, even on large disks, while it can take some hours with the traditional *fsck* (file system check) of the ext2 file system. For that reason, journaling is mainly important where after an unclean shutdown the system must be back on track as soon as possible, but can save quite a bit of time even for the desktop user.

For the above consideration, all journaled file systems are roughly equivalent. What distinguish ext3 is its compatibility with ext2: it uses the same metadata structure as ext2. Roughly speaking, ext3 is ext2 with the addition of a journal. This is a big advantage: if two file systems use different layouts of metadata, it is impossible to migrate a partition from the one to the other without losing the data (which means that you must go through the process of a back-up/restore); in our case, we can convert an existing ext2 file system to ext3 without losing data, in a few easy steps. Moreover, being built on top of ext2, ext3 can benefit from the impressive stability of ext2 and of the excellent recovery tools available for that file system. Another advantage is that ext3 has three types of journaling, and on the basis of which one you choose it can also guarantee the integrity of the data along with the consistency of the file system.

System requirements

It is reasonably easy to migrate from ext2 to ext3, provided that you have (or are going to gain) some confidence with upgrading system utilities and recompiling the kernel.

In fact, before you start the conversion, you must check that the ext2 tools you have in your system support ext3. You should be on the safe side if you run a recent distribution (RH 7.2 and Mandrake 8.1 offer ext3 as an option during the install process).

You should have a recent *util-linux* — in my system for instance I have version 2.11h; to check, give the command:

```
# /sbin/fdisk -v
```

and the answer will be something like

```
fdisk v2.11h
```

— and it is recommended that you install version 1.25 of *e2fsprogs*. To verify its version number:

```
# /sbin/fsck
```

```
fsck 1.25 (20-Sep-2001)
```

No devices specified to be checked!

In addition, you need at least *mount* version 2.11f.

```
# mount -V
```

```
mount: mount-2.11h
```

Warning: the above utilities are fundamental for the Linux OS, so be carefull when upgrading them. If possible we recommend you use precompiled packages specific to your distribution.

The kernel site

Linux has officially supported ext3 since 2.4.15 (which is unusable, by the way), while in the -ac series ext3 was supported even earlier. If you have a kernel with support for ext3, you should check that it is compiled either statically or as a module. If this is the case, you can move on to the next section. Otherwise the kernel has to be recompiled, and below we point out the relevant options to be enabled. Support for ext3 can also be enabled on less-recent kernels, using a patch available at <http://www.zip.com.au/~akpm/linux/ext3>

The procedure to patch the kernel is the following. As root:

```
# cd /usr/src/linux
```

```
# gunzip /path/to/patch/ext3-2.4-0.x.y.patch.gz | patch -p1
```

Assuming that you use *xconfig* to configure the kernel, the following options must be enabled (see screenshot):

Code maturity level options -> Prompt for development and/or incomplete code/drivers -> **Y**

File systems -> Ext3 journaling file system support (EXPERIMENTAL) -> **Y** or **M**

File systems -> JDB (ext3) debugging support -> **Y**

(The last option is not really necessary, but may be useful.)

You can now proceed to the compilation and the installation of the new kernel image in the usual way (refer to the kernel documentation for further details).

If you are planning to mount your root partition as ext3, you may want the ext3 support built into the kernel, otherwise you can have it as a module. You can still have it as a module if your / is ext3, but in this case you need to create an initial ram disk after you have converted the partition, directly (see the file *Documentation/initrd.txt* under the kernel source directory) or (recommended) via the *mkinitrd* command. On my system:

```
# mkinitrd initrd-2.4.8-26mdk.img 2.4.8-26mdk
```

```
mke2fs 1.25 (20-Sep-2001)
```

For more information

Andrew Morton's ext3 on 2.4 kernels usage page:

<http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html>

A good step-by-step guide

<http://ftp.uninett.no/pub/linux/packages/ext3/HOWTO/index.html>

A great (French) guide:

<http://qliu.free.fr/Linux/ext3.html>

The first of two excellent articles about ext3 by Daniel Robbins:

<http://www-106.ibm.com/developerworks/linux/library/l-fs7>

A Red Hat white paper on

ext3: <http://www.redhat.com/support/wpapers/redhat/ext3>

```
# mv initrd-2.4.8-26mdk.img /boot
```

Don't forget to modify the boot loader configuration file in such a way that it is aware of the ramdisk (and to re-run it in the case of *Lilo*). For instance, if you use *Lilo* the relevant line to add in the above example is

```
initrd=/boot/initrd-2.4.8-26mdk.img
```

in the section corresponding to the new kernel.

As usual, keep the old kernel image: it will be useful if you run into troubles even if you have only *ext3* file systems: a kernel without *ext3* support is still capable of mounting clean *ext3* partitions as *ext2*. Though not really difficult in principle, the building of a working kernel image can be a tricky operation: we advise inexperienced readers to carefully study the corresponding documentation and to learn how to recover a broken system before attempting at it.

Migration from ext2 to ext3

Once all the required software has been installed and the relevant kernel options have been enabled, it is really child's play to create an *ext3* file system on a spare empty partition. First of all, login as root, then if */dev/hda6* is your spare partition, just issue the command

```
# mke2fs -j /dev/hda6
```

To convert an existing partition with data on it that you wish to keep, the command is

```
# tune2fs -j /dev/hda6
```

This will add a file named *journal* (the actual journal) at the root of the partition. The output should look similar to this:

```
tune2fs 1.25 (20-Sep-2001)
```

```
Creating journal inode: done
```

```
This filesystem will be automatically checked every 35 mounts
or 180 days, whichever comes first. Use tune2fs -c or -i to
override.
```

(we will explain the last two lines in a while).

We can check the type of the filesystem with the command *dumpe2fs*:

```
# dumpe2fs /dev/hda6 | more
```

```
Filesystem features: has_journal filetype sparse_super
```

We can now remount our partition as *ext3*. Suppose it is mounted under */opt*. Then, by typing

```
# umount /dev/hda6 && mount -t ext3 /dev/hda6 /opt
```

we can quickly start enjoying the benefits of *ext3*. The last line of the output of *mount -v* will convince us that */dev/hda6* has been mounted as *ext3*, as expected:

```
# mount -v //command output//
```

```
...
```

```
/dev/hda6 on /opt type ext3 (rw)
```

To allow the kernel to mount this file system as *ext3* at boot time, the corresponding line in */etc/fstab* must be edited, and “*ext2*” must be replaced by “*ext3*”. For instance, the following line

```
/dev/hda6 /opt ext2 defaults 1 2
```

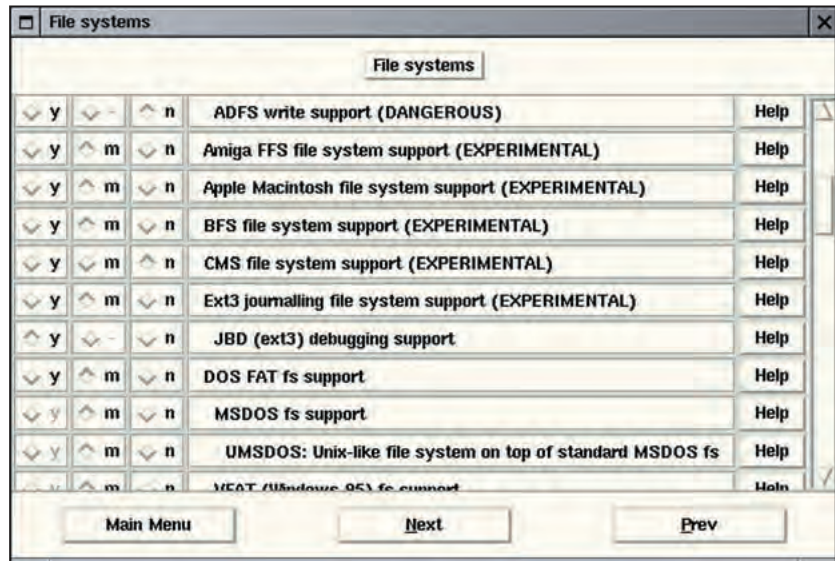
must be modified into

```
/dev/hda6 /opt ext3 defaults 1 2
```

(if instead of “*ext3*” in the original file there is “*auto*” the kernel will automatically realise that the filesystem is *ext3* and mount it as *ext3*). Of course for a new partition the appropriate entry in */etc/fstab* must be created.

This process can be repeated for all the file systems you want to migrate, included the root partition (in this case you need to reboot the computer before remounting it, and you should remember to create the initial ram disk as described above if *ext3* support is compiled as a module in the kernel).

Ext3 has three different journaling modes. The mode



“*data=writeback*” will just guarantee the integrity of the file system. “*data=ordered*”, the default mode, ensures some integrity of the data, while there is a third mode, “*data=journal*” in which both data and metadata are journaled. The best mode depends on the application: of course the more you journal, the slower the writing process. Anyway, to overwrite the default journaling mode, the modification to the above *fstab* line reads

```
/dev/hda6 /opt ext3 defaults,data=journal 1 2
```


The big advantage of *ext3* is that you are almost always guaranteed the consistency of the file system. Nevertheless by default *fsck* will be run after 35 mounts of a partition or after 180 days, according to the output of the *tune2fs -j* command above. To avoid that, you can run

```
# tune2fs -c 0 -i 0 /dev/hda6
```

But remember that “almost always” is not “always”: a broken kernel or a faulty hard disk can corrupt your file system, and the tedious thing is that it might still be marked as “clean”. So if you choose the above settings, it will be your responsibility to schedule routine checks of the filesystem.

Conclusions

There are several reasons why a journaled file system may be seen as preferable to the traditional *ext2*. Among these journaled file systems *ext3* offers perfect compatibility with *ext2* and an easy migration process.

Indeed, with its three journaling modes that work with almost no loss of performance in most cases, *ext3* should suit everyone, from the desktop to the professional user. 

The options to be enabled in the kernel configuration in order to have *ext3* supported via modules.

What if I want to go back?

It may make you feel a bit more comfortable to know that a clean *ext3* filesystem can always be mounted as *ext2*. For instance, if for some reason you want to mount an *ext3* root file system as *ext2* at boot, you can pass the *rootfstype=ext2* parameter to *Lilo*:

```
LILO: linux rootfstype=ext2
```

What's more, the migration process is reversible. Should you want to migrate back (but we bet you won't), in the example discussed through the article this can be accomplished as follows. First of all, we unmount the partition:

```
# umount /dev/hda6
```

Then, we remove the **has journal** feature:

```
# tune2fs -O ^has_journal /dev/hda6
and suppress the inode of the journal:
```

```
# fsck /dev/hda6
```

After that, */dev/hda6* is now remounted as *ext2*

```
# mount -t ext2 /dev/hda6 /opt
```

Remove the “immutable” attribute from the file *journal*

```
# chattr -i /opt/journal
```

Finally we remove the journal

```
# rm /opt/journal
```


LinuxFormatTutorialPartitionImaging

BACKUPS MADE EASY

Backing up partitions

Backing up isn't hard to do with **Nick Veitch's** guide to partition imaging, so go and save your data!

We have often gone on at length about the necessity of keeping backups, as protection against unforeseen events. You don't realise how valuable your data is until you miss it!

Backing up files is one thing, but it is often more useful to backup entire partitions. Restoring the root partition of a drive from a file backup can be tricky, and what do you do if the host system has no viable installed OS?

The solution is to backup entire partition images (although file backups are also useful). There are further advantages to this approach. If you have a set of identical client machines (in a web café environment for example), you can simply have one backup image which can be used to restore any of the clients. This can also make upgrading clients much easier — install and configure the software correctly on one, and you can then use an image to update all the others. With the right software, you could even do this over a network, simultaneously.

Using the dd tool

For simple partition backups, you might have used the *dd* tool. This versatile tool is used for copying data blocks from one device to another or to a file. For example, to back up your /boot partition which lives on /dev/hda1 you might use:

```
dd if=/dev/hda1 of=$HOME/boot.img
```

Which quite simply copies the blocks of /dev/hda1 until it reaches the end. It is possible to specify further options for *dd* of course, but these often aren't useful for copying partitions.

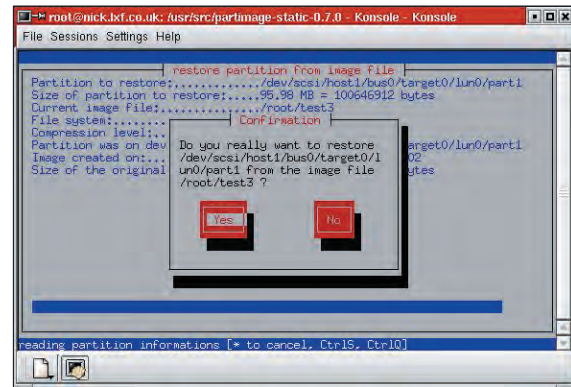
While *dd* is perfectly useful for small images like floppy disks, boot disks and your /boot partition, it isn't quite so useful when it comes to larger partitions. If you want to duplicate, e.g., a 2GB partition, you are going to need another 2GB of filesystem space somewhere else. This is still the case even if only a small portion of the partition is actually in use. You can retroactively compress the image file, but you still need the disk space to begin with.

Enter Partition Image

These are the kind of limitations that *Partition Image* was developed to overcome. It can dramatically reduce the time and space required to complete partition backups in two ways:

- It only backs up used blocks in the partition. This alone is usually a major space saver. Few people run their partitions to 100% full, so there is space to be saved here.
- The software can also use a variety of compression tools to reduce the size of the image being created.

For all but the very simplest of image files, using *Partition*



Restoring a partition should be straightforward, and quicker than the original backup.

Compression libraries

libz	gzip compression library	www.zlib.org
libbz2	bzip2 compression	http://sources.redhat.com/bzip2/
liblz	fast compression library	http://www.oberhumer.com/opensource/lzo/
libnewt	curses-like GUI system	ftp://ftp.redhat.com/pub/redhat/linux/code/newt/
libslang	slang, required for newt	http://www.s-lang.org
libssl	OpenSSL	http://www.openssl.org
libmcrypt	Mcrypt for secure networking	http://mcrypt.hellug.gr

Image is much more effective, both from a time and storage point of view. The time penalty incurred by compressing the data is often made up for by the fact that you are compressing a smaller file, and the subsequent reduced time taken to copy the image onto backup devices where necessary.

The story so far

Francois Dupoux and Franck Ladurelle released version 0.2 of *Partition Image* in May of 2000. Since then it has grown from its fairly basic beginnings to support more and more filesystems, additional compression formats, and an easy to use GUI, based on the curses-like *newt* system.

You can get *PI* from the homepage at (www.partitionimage.org), or from the files on this month's DVD and CD. There are a number of ways to install *PI*, and a number of packages available. If you just want to run the latest stable version, you are probably best off downloading the static binaries.

If you want to compile it yourself either because there isn't a binary available or you the latest CVS version, there are a few libraries you will need — see the *Compression libraries* box.

To compile from source (once have installed the libraries), you can follow the usual procedure of running **configure** and **make**. However, you may want to configure with options for additional filesystem support, or to control the location of SSL (or to disable SSL on the server. Running **./configure --help** will give you a full list of the options. So for example, you might want to say

```
./configure --enable-xfs
```

```
make
```

```
make install
```

If you have enabled SSL, you'll want to make the certificates:

```
make certificates
```

Partition tables

Before we continue to talk about backing up partitions, we need to discuss how to back up partition tables too.

Networking

As mentioned, it is possible to set *Partition Image* up to work over a network. Then you can have a central server with your image files, making it much easier to centrally control the data that you backup and restore.

The *partimaged* program runs the server, and all you have to do is open a shell and run it. The user 'partimag' should exist on your server, so you have to create this before you run the server for the first time.

A screen will open showing the status of the 10 allowed connections, and that's it — the server is ready for operation. Remember to make a note of the IP number if the server's hostname isn't included on DNS.

Bear in mind that the limiting factor of backing up over the network is liable to be your network speed. Backing up a 10GB partition over a 10Mb line may take more than a few hours.

LinuxFormatTutorialPartitionImaging

The partition table is the information stored on the drive which tells the OS where, what type and how many partitions are on the drive. As you'll probably know if you've ever partitioned a disk, there are all sorts of ways to divide one up. Without the information on how a disk is partitioned, it's going to be a lot harder to restore images if you need to.

The *Partition Image* software itself does not (yet) save the partition information, so you'll have to save this yourself.

For a disk which has a few primary partitions, this is not too difficult. The original DOS specification for partitions allowed for up to four partitions on a single device. Under Linux these would be, for example, hda1, hda2, hda3, hda4. These types of partitions are known as primary partitions, and every disk will have at least one. The information describing these partitions is included in the MBR (Master Boot Record) of the device, which is always contained in the first 512 bytes of the disk.

But having only 4 partitions became a bit of a limitation, especially when much larger disks came along. The solution was to create logical partitions "inside" one of the primary partitions. This works by allocating a primary partition, then chaining the data for the logical partitions inside them. The first logical partition points to the location of the next and so on. That's why you may often find a disk with partitions labelled hda1, hda5, hda6, hda7, but with no hda2. The numbers 1-4 are reserved for the primary partitions.

Having these linked partitions is a bit of a pain because there is no single part of the disk to read the data from. So in order to save the partition info for the disk, we need to do two things:

- 1 Backup the Master Boot Record
- 2 Backup the information for any logical partitions.

There are one or two utilities to help with this. If you are using Mandrake 8.x, you can use the *diskdrake* utility to store the partition information to a file for you. (Select the drive, and click on the "More" button in the bottom of the Window). This stores the data in a fairly human readable form too.

The alternative is to backup the data using standard Linux tools. The first task is easily accomplished using *dd*. As we are only talking about a 512 byte file, it's more than up to the task. For example, to save the MBR from your first IDE hard drive:

```
dd if=/dev/hda of=hda.mbr bs=512 count=1
```

To backup logical partitions, you can use *sfdisk*, one of the *util-linux* tools packages, and will be installed as part of almost any distro. If you don't have it, you can get *util-linux* from your local kernel mirror or the main www.kernel.org site.

You should be careful with *sfdisk*, as it is possible to lose data using it. Check out the man pages for the options if you are unsure. To backup the partition information, use:

```
sfdisk -d /dev/hda > hda.sfd
```

This will dump the partition info into the file *hda.sfd*. Armed with these two files, you should be able to reconstruct the

Compression methods comparison

type	time taken	size
No compression	142s	81MB
lzo compression	144s	43MB
gzip compression	146s	41MB
bzip2 compression	155s	38MB

These simple tests serve as a guide only. They were performed on a PIII/850MHz machine with standard IDE hard drives. The differences in time taken and final size will differ depending on the type of data stored in the partition, and the speed of your processor.

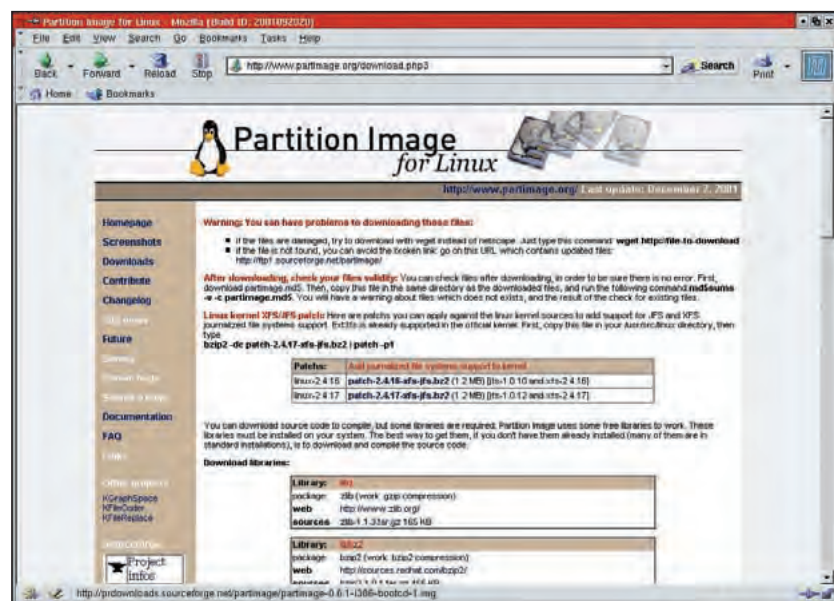
Hard drives

Before performing a disk intensive task like backing up or restoring a partition, it might be a wise idea to make sure that your drives are configured most efficiently. Normally, if you just install a Linux distro and don't check the settings, things will be set to the lowest common denominator. This works, but doesn't get the most out of your hardware.

The first and most basic thing to check is that your system is using an appropriate bus speed. You'll probably find that by default your IDE bus Speed is set to 33MHz, which is a safe

value for PCI/IDE systems, but you may want to change this. The bus speed setting can dramatically effect the timings for pio modes if enabled on your drives.

You should also check whether your drives are using PIO modes, DMA or whatever. The *hdparm* utility can be used for setting these modes, (it's part of the *util-linux* package you can find on the kernel website and it's mirrors). However, you can really screw up your disk using this, so read the documentation! In doubt, best stick to slower backups...



partition table of any disk. So for goodness sake, copy these files somewhere safe. Please don't leave them on your hard disk — they won't be much use to you there if that's the disk or partition that has a problem! Best to keep them on a floppy (or a couple of floppies for extra security) or better still, burn them to the bootable CD you're going to make later...

The website contains up to date information, usage guides, and many different versions of the software.

Backing up a partition

Before we start backing up a partition, you should first know where the partition is. *Partition Image* uses devfs-style partition labels, so if you are backing up hda1 for example, it will be listed as ide/host0/bus0/target0/lun0/part1. For hdb1, the listing would be ide/host0/bus0/target1/lun0/part1. There is a little help in that *Partition Image* will also display the filesystem installed and the partition size. Remember that the partition should really be unmounted before backup, to ensure that no files are open or can change during the backup.

Restoring partition information

If your partition information has been lost, or you want to duplicate the partitions from an identical drive, then we can restore the information we saved earlier:

```
dd if=hda.mbr of=/dev/hda bs=512 count=1
```

```
sfdisk -O changelog.sfd /dev/hda <hda.sfd
```

IMPORTANT: This will rewrite your partition table. Make sure that you type the commands correctly, and use the correct backup files. Incorrect partition data could result in loss of data.



LinuxFormatTutorialPartitionImaging

« The *sfdisk* utility has safeguards to make it hard to write partition data to a mounted drive. You should unmount the drive before you write the partition table. Unfortunate things can happen if you change the partition info when files are being accessed on the disk! If you use the wrong file, or discover a mistake, you can usually recover the partition with the aid of the changelog file we created when we were writing the *sfdisk* data:

```
sfdisk -l changelog.sfd
```

Of course, if you have used *Diskdrake* to back up your partition information, you can also use that to re-write the data.

Whichever method you use, you will normally need a reboot for the changes to be noticed by Linux, so do that before attempting to restore any partitions.

Restoring the partition

Once your partition tables are sound, you can begin restoring the partition. This is even easier than backing up, as there are

less options to worry about. Select the partition you want to restore from the list (choose carefully) and enter the filename of the file containing the data. If you are restoring over a network, you'll have to select the IP or hostname of the server too.

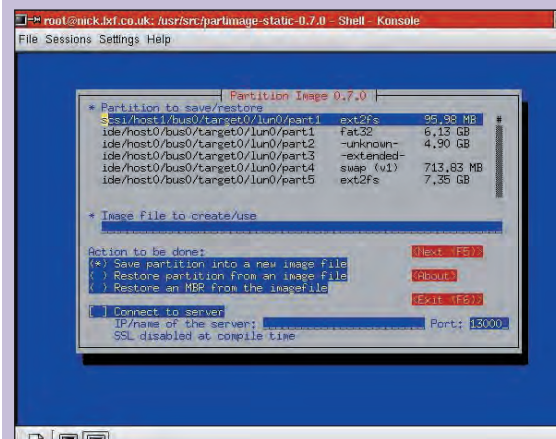
Press F5 and, if you entered a description for the saved file, this text will now be displayed. You can 'simulate' the write, which at least will check for potential problems with the image file (if it has become corrupt or it's the wrong size). You can also choose to erase empty blocks with zero values. This will overwrite any data in the partition and make it all but impossible to retrieve any of the data from the drive.

Conclusion

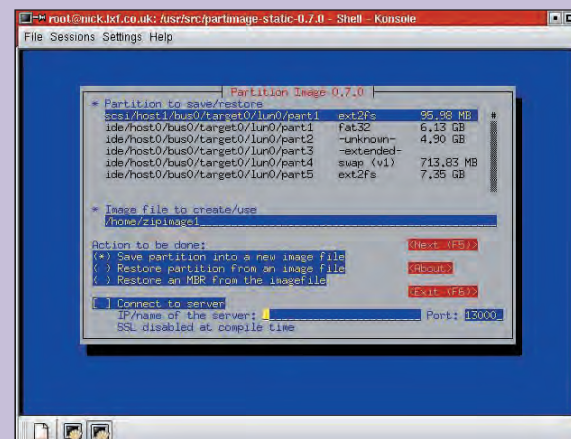
While not exactly fun, *Partition Image* certainly makes the backing up of partitions fairly straightforward and simple. Nobody likes backing up data, but backing up complete partitions offers an easy and convenient solution for those who find it necessary. If you have all your partitions backed up, what's the worst that can happen?

Partition Image step-by-step

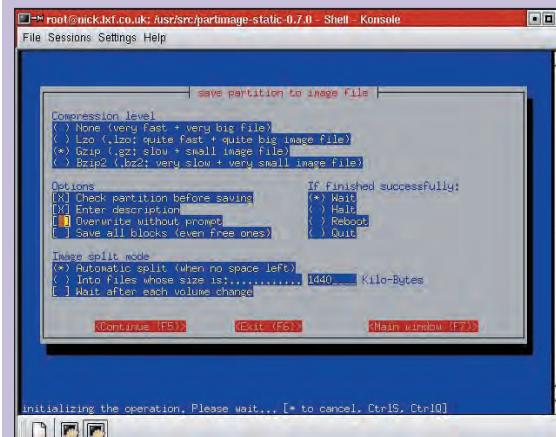
We walk you through the backup procedure



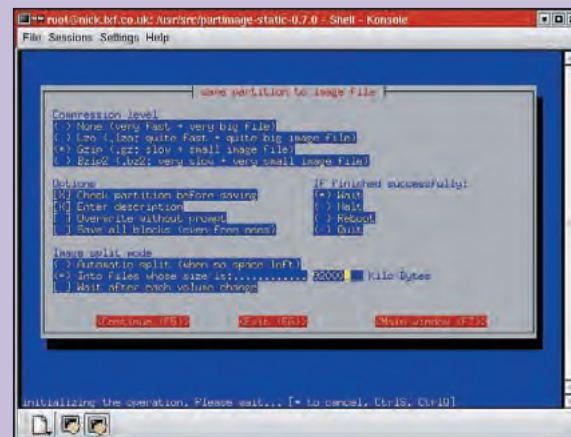
STEP 1 Run the *Partition Image* program from the console. Make it a reasonable sized window so you can see what's going on! You'll now see a curses-style display of the main menu. The first detected partition will be highlighted. Use the cursor arrows to highlight the partition you wish to back up.



STEP 2 Press the TAB key to move between the options. Type in the pathname of the file to store the partition data in. You can leave the rest of the options for the moment, as we are not connecting to a server, nor are we performing any restore options (the default is set to backup partitions, as indicated by the asterisk).



STEP 3 Press f5 to continue: On the next screen we can set the backup options. The first bank of options selects the compression level. The default options are usually fine. If you choose to check the partition (a good idea) you'll soon find out whether you remembered to unmount it first.



STEP 4 If you want to split the files to fit on CDs or removable disks, you should check these options here. The Wait after each volume change can be useful if you are writing directly to removable media. In the example here we have specified a size suitable for fitting on Zip disks.

LinuxFormatTutorialPartitionImaging

Creating a boot CD for x86

Also on our disks this issue, and available from the *Partition Image* website, is a useful El Torito image for creating a bootable CD. There are a number of potential uses for this:

- Backing up your root partition (which you won't be able to do when it's mounted)

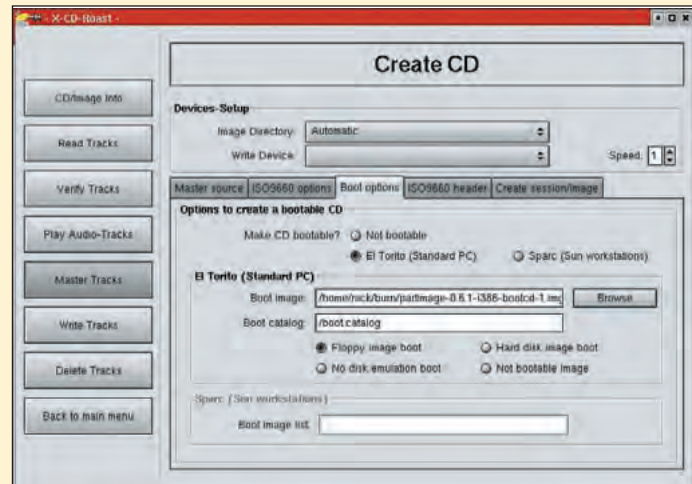
- Backing up a partition from a non-linux machine (i.e. a Windows desktop)
- Restoring partitions to a machine with no viable OS

Creation of the boot disk is simple. You just have to copy the El Torito image to a directory and master a CD, using this file as the El Torito image (For various

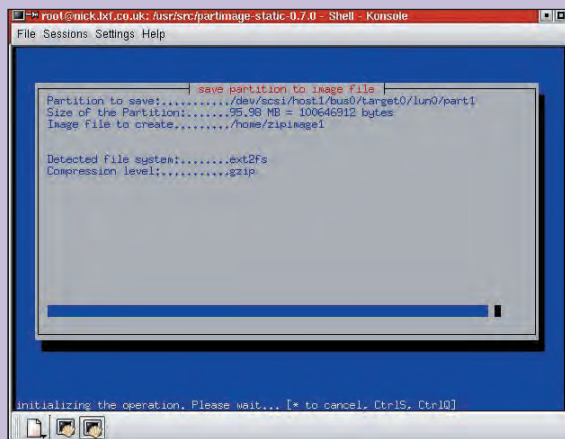
reasons, this is the name given to the file which is used as the bootable portion of a CDROM).

If you have a generic partition image, you can burn this to the CD too, enabling you to boot and restore your system without access to any other drives. As the El Torito image takes up a negligible amount of space on the CD, there is 650MB, more or less, of space for any partition backups.

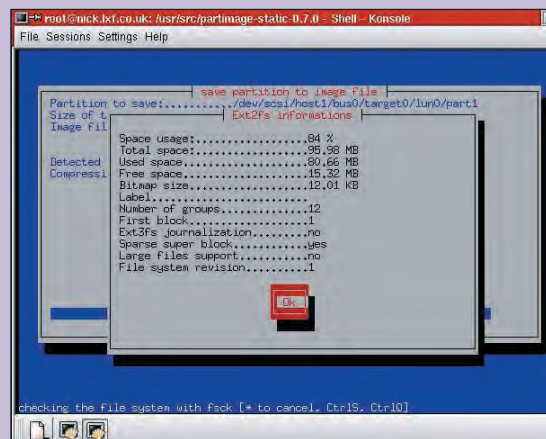
You can create an image and use it with the standard Linux tools, *mkisofs* and *cdrrecord*, but as GUIs like *XCDRoast* support El Torito information now, it's probably easier to use this.



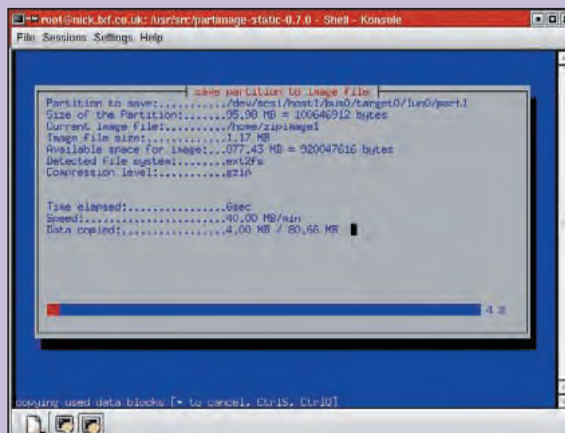
Point *XCDroast* at the El Torito image to create a bootable CD.



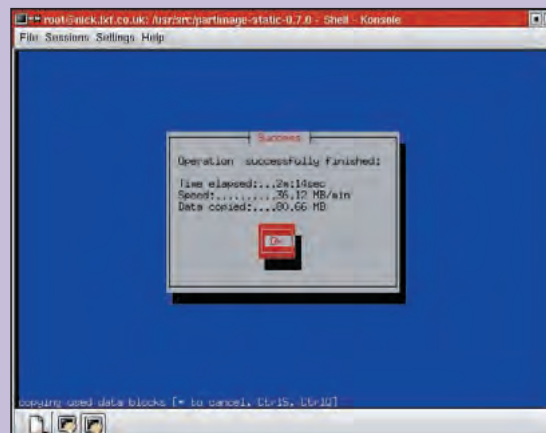
STEP 5 Hit F5 and you will have the option to enter a useful description. Do so and then the operation will initialise. This is the point at which the filesystem is checked (if you chose that option) so it can take a few moments. Wait patiently.




STEP 6 An info screen will pop up to tell you about the partition you are backing up. This may have various pieces of data, depending on the filesystem in use on that partition, in addition to some info about blocks & the size and used space on the partition. Press ENTER to continue.



STEP 7 The operation will now start, giving you a running total of the amount of data stored so far, and various other statistics to keep you happy. The time taken for saving the image obviously depends a great deal on how much data you are backing up!



STEP 8 When the operation has finished, you'll see this screen showing you what exactly has happened (if you have chosen to wait for split files, you'll get messages in between for media changing). Click on Okay and you are done. 



QUICK AND DIRTY OOP

Practical Python

In a one off tutorial, **Nick Veitch** explains how quick and dirty Python can get the job done

First of all, I feel we should explain ourselves here. The code contained in this feature is not the ideal example of that language and its usage. It wasn't constructed as a teaching exercise, but in response to a real need.

Let me explain. Towards the end of 2001 we were planning to run several surveys from the *LinuxFormat* website. The main Futurenet site runs surveys all the time, and has various proprietary solutions for constructing them and handling the data. Being rather single-minded, we decided we could run it on our own server, create our own forms, and use our own database.

It is, in theory, quite straightforward. All you need is some HTML with a form, a script to post that data, and a database to post it to. However, the number of people involved in creating, checking, re-wording and otherwise altering the questions and what data was to be collected meant that it became impossible to hand-code the form (well, not impossible, but certainly very tedious and time consuming). What we really needed was a system to generate the form from a list of questions. That way we could easily change the questions and just process the list. And of course, should we ever want to do another survey, the job would be a lot easier.

Time was spent searching the usual archives of knowledge for some Linux-based script that would do the job. Several candidates were investigated, but none of them seemed to fit the bill – mostly because they wanted to be too complete, often because they were over elaborate. So we decided to write our own, which is why I am also writing this brief tutorial. Rather than let the script lurk on our hard disks in case it should be needed again, we decided it may be of use to someone, at the least as an example of how you can easily knock together a worthwhile and useable script in python, even if you don't know it so well.

Before we proceed though, some caveats to our code:

- It is dirty. It was designed to do a job quickly. There are more elegant ways to, for example, construct the loop which checks off the type of form element. However, you don't need to write elegant code to get a job done.
- There are plenty of ways it can be improved.
- It hasn't been idiot-proofed. If your input file is junk, it will most likely throw out an error. It probably won't construct a bad form, but that's for you to check.
- The HTML it outputs is correct, but not nicely formatted – a lot of the output strings are constructed in the interests of keeping the code manageable rather than making the output pretty.
- The output is a full HTML page, but you probably won't want to use it straight as it comes – however, it is ideal as the basis of a page design.

The first step was to come up with the format of the input file. It needed to be easy to understand, easy to edit and easy to process. And easy to add stuff to without rewriting everything. The first solution, and the one we stuck with, was to use a CSV (comma separated values) file. As it is essentially just text, it suited perfectly. The format is:

Number, type, question text, option1, option2, option3, ...

The number wasn't really necessary, but it was just as easy to include the question number separate from the text. As this is the way we started, it was easier to keep it this way. Now all we had to do was determine the type of questions. Radio buttons, check boxes and lists, as well as text entry, were to be used. so the type field just contains "radio", "check", "list".

Subsequent types were added to this spec, which were easy to accommodate using this non-rigid format – all we had to do was check for a different string in the 'type' field. Indeed, additional types made use of the option fields in different ways. An array of radio buttons was needed for some questions, so 'Option 1' contained the number of columns, then the next however many fields contained the column headers, followed by the rows.

Later types such as a comment field and an email field were also added – it's good to keep your options open until you know where you are going. The Python script we wrote to handle this data evolved with the changing format. Because we use modular functions in the script, it was easy to expand it with the developing format of the data files.

The first script

As you can see from the listing, the script starts off by importing useful python modules. The string module is a necessity for this type of project, and the sys module gives us file access.

Next we have the function definitions, most of which deal with the types of files and outputting the right HTML for each. Skip forward to the main program code, and you'll find it's quite short.

Open the files we need (the datafile containing the data, the HTML file to output and another we'll come onto in a minute) and output the headers for the HTML, and then you can go on to process the list.

Python's ability with lists, strings and files makes this a doddle:

```
for line in datafile.readlines();
```

sets up the loop for us, in one easy statement. **readlines()** is a function of the file class, which returns the next line of the file each time it is called. The rest of the statement assigns the variable 'line' to this value, until there are no more values to assign (when the file has reached the end). So that one line of python code sorts out pretty much all of the file input for you.

The line variable is then easily split into its individual components using the string class's **split** function, and using the fact that each value is comma delimited. List is not just a variable, but a set of string variables (or a *tuple*) which is handled like an

array. There is no need to predefine the parameters of this tuple, python handles it automatically. The tuple can be accessed using an index, just like an array, so immediately we can access the specific variables in each line. The **string.strip** function clears out any untidy spaces that might exist in the original input file.

A bank of **if** statements determines what type of question we are looking at, and passes pointers to the relevant functions. These can then use the filehandle, and the list to generate the HTML.

Each function is slightly different, but they merely output HTML text depending on the values specified for the questions. No rocket science there. The only complicated one is the multiradio type, which creates a table array of inputs with a nested loop, but even that isn't incomprehensible.

Making the database

Earlier on we talked about creating the structure of the database. Well, when we have a list, we can use that to create the table data. In fact we use the secondary output from the first script. We aren't worried about the structure of the questions, just the values that will be passed. For later identification, we'll use the 'safe' labels as field labels for the table too.

As you can probably see, the second script is just an ammended version of the first. It would easily be possible to


merge this into the first script, by writing the 'survey-submit.txt' to a temp file and reading it in again. You might want to use a different type of data capture though, so we have left this as two scripts. The resulting survey.SQL file is a valid SQL instruction, so you can set up a database easily in MySQL:

```
mysql
> create survey
> \u survey
> \. survey.SQL
```

The missing link between the HTML form and the database is a cgi script. What you end up using depends on your server, but a simple PHP script could do the job for you (one is included on the CD). There is an example CSV file on the CD for you, too.

Conclusion

So there you go. It might not be the prettiest code, it might not be very robust. But in it's favour it works, and the whole system was written in less than a day (a whole lot less time than it took to sort out what the questions were going to be, believe me).

We could have written this in Perl, but it would probably be fairly indecipherable by now. As it is, there are elements of this code which could easily be recycled for other projects, or as part of a more elaborate system. 

listtoform.py

```
#!/usr/bin/python

import string, sys

#####
# Safename removes dodgy characters
# from field names
#####
def safename(namestring):
    namestring = string.replace(namestring, '/', '');
    namestring = string.replace(namestring, '\\', '_');
    namestring = string.replace(namestring, '&', '');
    namestring = string.replace(namestring, '+', '_');
    namestring = string.replace(namestring, '=', '');
    namestring = string.replace(namestring, '!', '');
    namestring = string.replace(namestring, '-', '_');
    return(namestring);

#####
# Define functions for handling text types
#####

def typetext(list,entrycount,z,p):
    print 'text function initialised';
    name = safename(list[0]+'text');
    p.write(name+'.text\n');
    z.write('\n<!-- Text Input field -->\n');
    z.write('<br><b>Q'+list[0]+' '+list[2]+'</b>');
    z.write('<input type="Text" name="'+name+'" value=""><br>');
    return entrycount+1;

#####
# Email type - different from text, because
# this gets checked. Must be unique
#####
def typeemail(list,entrycount,z,p):
    print 'email function initialised';

    name = 'email';
    p.write(name+'.text\n');
    z.write('\n<!-- Email Text Input field -->\n');
    z.write('<br><b>Q'+list[0]+' '+list[2]+'</b>');
    z.write('<input type="Text" name="'+name+'" value=""><br>');
    return entrycount+1;

#####
# Message type
# For inserting HTML or instructions
#####
def typemsg(list,entrycount,z,p):
    print 'message function initialised';
    z.write('\n\n<P>'+list[2]+'</P>\n\n');
    return entrycount;

#####
# A radio button
#####

def typeradio(list,entrycount,z,p):
    print 'radio function initialised';
    print 'list one is',list[0];
    print 'list question is',list[2];
    z.write('\n<!-- New Radio Button Question -->\n');
    z.write('<table>\n<tr>\n<td colspan=2><b>Q'+list[0]+' '+list[2]
+ '</b><br>\n');
    z.write(' </td>\n </tr>');
    z.write(' <tr>\n<td align="left">\n');
    name=safename(list[0]+'radio');
    i=3;
    while i < len(list):
        if list[i] is "":
            break;
        value = i-2;
        # Numeric values are better for processing, but you might want text,
```



LinuxFormatTutorialPython

listtoform.py continued

```
# in which case use this:
#         value=string.replace(list[i],',','');
# and change the '+str(value)+' to '+value+' here
        z.write('<input type="radio" name="'+name+'" value=
'+str(value)+'"> '+list[i]+'\\n');
        i=i+1;
        z.write(' </td>\\n </tr>');
        z.write('<tr><td></td></tr>\\n</table>');
        p.write(name+',radio\\n');
        return entrycount+1;
#####
# Multiple Radio button
# list[3] contains number of columns
#####
def typemultiradio(list,entrycount,z,p):
    print 'multiradio function initialised';

    z.write('\\n<!-- New Multiple Radio Button Question -->\\n');
    z.write('<table width =550>\\n<tr>\\n<td colspan=5><b>Q'+list[0]+'
'+list[2]+'</b><br>\\n');
    z.write(' </td>\\n </tr>');

    #name=list[0]+'radio';
    i=int(list[3]); # number of columns
    columns=i;
    perwidth=100/(columns+1);
    z.write(' <tr><td width="'+str(perwidth)+'"></td>\\n');
    j=1;
    while j < i+1:
        z.write('<td width="'+str(perwidth)+'">
<h4>'+list[j+3]+'</h4></td>');
        j=j+1;
        z.write('</tr>\\n');
        i=i+4;
        while i < len(list):
            if list[i] is ":
                break;
            z.write('<tr>\\n<td>'+list[i]+'</td>\\n');
            name = safename(list[0]+'multiradio_' +str(i-3-int(list[3])));
            p.write(name+',radio\\n');
            j=0;
            while j < int(list[3]):
                z.write('<td align="left" width="'+str(perwidth)+'%">');
                z.write('<input type="radio" name="'+name+'" value
="'+str(j+1)+'">\\n');
                j=j+1;
                z.write('</tr>\\n');
                i=i+1;
            z.write(' \\n ');
            z.write('<tr><td></td></tr>\\n</table>')
            return entrycount+1+j;

#####
# A checkbox type question
#####
def typecheck(list,entrycount,z,p):
    print 'check function initialised';
    z.write('\\n\\n<!-- New Check Button Question -->\\n\\n');

        z.write('<table>\\n<tr>\\n<td colspan=2><b>Q'+list[0]+' '+list[2]+'
'</b><br>\\n');
        z.write(' </td>\\n </tr>');
        z.write('<tr>\\n<td align="left">\\n');
        i=3;
        while i < len(list):
            if list[i] is ":
                break;
            name=safename(list[0]+'check_' +str(i-2));
            name = string.replace(name,',','');
            p.write(name+',check\\n');
            z.write('<input type="Checkbox" name="'+name+'" value="1"> '
+list[i]+'<br>\\n');
            #NB value of 1 is given to a checked box
            i=i+1;
            z.write('\\n</table>');
            return entrycount+i-3;

#####
# A list type question
#####
def typelist(list,entrycount,z,p):
    print 'list function initialised';
    z.write('\\n<!-- New List Button Question -->\\n');
    z.write('<table>\\n<tr>\\n<td colspan=5><b>Q'+list[0]+' '+list[2]+'
'</b><br>\\n');
    z.write(' </td></tr>\\n');
    z.write(' <tr>\\n');
    name=safename(list[0]+'list');
    p.write(name+',list\\n');
    i=3;
    z.write('<td><select name="'+name+'">\\n');
    z.write('<t<option> ----- </option>');
    while i < len(list):
        if list[i] is ":
            break;
        z.write('<t<option> '+list[i]+'</option>\\n');
        i=i+1;
    z.write('</select>\\n</td>\\n</tr>');
    z.write('<tr><td></td></tr>\\n</table>');
    return entrycount+1;

#####
# Add mailing list boxes
#####
def mailingstuff(z,p):
    p.write("listLXF,check\\n");
    p.write("listother,check\\n");
    z.write("<br><HR>\\n<P>Your email address is only used for
verification purposes, and will not be added\\n");
    z.write("to any mailing lists without your knowledge. However, please
tick the following\\n");
    z.write("If you would like to receive information from Linux Format or
the sponsors of this survey</P>\\n");
    z.write('<br><input type="Checkbox" name="listLXF" value="yes">
<H4>Please add me to the Linux Format mailing list</H4><br>\\n');
```

```

        z.write('<br><input type="Checkbox" name="listother" value="yes">
<H4>Please notify me of other Linux related offers and
information</H4><br><HR>\n');

#####
#
# MAIN CODE STARTS HERE
#
#####

#####
# Open the output files for HTML
# and the input files of CSV text
#####
z=open("form.html","w");
p=open("survey-submit.txt","w");
datafile=open("surveylist.csv","r");

#write HTML headers to file
z.write('<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
EN">\n');
z.write('<HTML>\n');
z.write('<head><title>Survey</title>\n');
z.write('<LINK REL=stylesheet HREF="survey.css" TYPE="text/css">\n');
z.write('</HEAD>\n');
z.write('<BODY bgcolor=python ="#FFFFFF" TEXT="#000011">\n');
z.write('<form action="survey-submit.php" method="POST" name=
"survey">');

entrycount=0;

#main loop -reads input line at a time
for line in datafile.readlines():
    list=string.split(line,');
    a = string.strip(string.lower(list[1]));
#check type

```

```

    if string.find(a,'radi') is 0:
        entrycount=typeradio(list,entrycount,z,p);
        print list[0],' radio - processed';
    if string.find(a,'chec') is 0:
        entrycount=typecheck(list,entrycount,z,p);
        print list[0],' check - processed';
    if string.find(a,'mult') is 0:
        entrycount=typemultiradio(list,entrycount,z,p);
        print list[0],' multiradio - processed';
    if string.find(a,'text') is 0:
        entrycount=typetext(list,entrycount,z,p);
        print list[0],' text - processed';
    if string.find(a,'list') is 0:
        entrycount=typelist(list,entrycount,z,p);
        print list[0],' list - processed';
    if string.find(a,'email') is 0:
        entrycount=typeemail(list,entrycount,z,p);
        print list[0],' email - processed';
    if string.find(a,'message') is 0:
        entrycount=typemsg(list,entrycount,z,p);
        print list[0],' message - processed';
mailingstuff(z,p);

#finish off the HTML
z.write('<P> Thankyou for your time in filling out this form. To submit the
information');
z.write(' press the Submit button below </P>');
z.write('<table><tr><td ><input type="submit"><input type="reset">
</td></tr></table>');
print 'entries =',entrycount;
z.write('</form></BODY></HTML>');

#close files nicely
z.close();
p.close();
datafile.close();

```

listtoSQL.py

```

#!/usr/bin/python

import string, sys, os, commands

#####
# Open the output files for HTML
# and the input files of CSV text
#####

datafile=open("survey-submit.txt","r")
p=open("survey.SQL","w")

p.write('CREATE TABLE submission (id int DEFAULT "0" not null
AUTO_INCREMENT ');
fields=''
for line in datafile.readlines():
    list=string.split(line,');
    z = string.strip(string.lower(list[1]));
    if string.find(z,'check') is 0:

```

```

        fieldtype=' tinyint(2) ';
        print list[0],' radio - processed';
    if string.find(z,'list') is 0:
        fieldtype=' varchar(30) ';
        print list[0],' check - processed';
    if string.find(z,'radio') is 0:
        fieldtype=' tinyint(2) ';
        print list[0],' check - processed';
    if string.find(z,'text') is 0:
        fieldtype=' varchar(30) ';
        print list[0],' check - processed';
    if string.find(z,'email') is 0:
        fieldtype=' varchar(30) ';
        print list[0],' check - processed';
    fields=fields+' '+list[0];
    fields=fields+' '+fieldtype;
p.write(fields+' , ip varchar(16) , PRIMARY KEY (id), UNIQUE id(id);');

p.close();

```


VERSION CONTROL SYSTEM

CVS client guide

PART 1 Too many cooks spoil the broth, but with CVS**Jono Bacon** shows you how many developers can make light work of coding without descending into confusion.

Welcome to a two part series on getting to grips with one of the most common tools in the developer's toolbox in modern software development – the *Concurrent Versioning System* (CVS). In this series I will be going through all the major features of CVS and how to get the most out of it; from using CVS on other people's CVS servers to setting up your own CVS server. We will begin in this issue discussing what CVS is, the point of it all and how to use it to get access to other people's CVS servers. We will then continue next month by looking at what is needed to build your own CVS server.

What is CVS?

CVS is a version control system. "What is a version control system?" I hear you cry out. Well, let me explain with an example. Imagine you are a maintainer of a project. You are in control of the entire project and how it runs. Let us now assume that after a recent announcement you made of the project, you get an email from two developers who would like to contribute some code to your project. This is great news and can increase the development pace of your application. The problem you have now is how you can manage all three developers working on the same project. One option could be for the other developers to send you patches (small files containing new code for a project) which you can then apply to your source code. This is fine until you realise that after every patch you will need to release the source code so the developers are up to date. Now, let us now assume that developer 1 writes a patch and developer 2 writes a different patch that uses some of the same source files. These two patches are fine by themselves but together there will be a

conflict. The conflict would have to be resolved by you, and it may be in a part of the code you are not very familiar with. As you can see, this is a very messy situation for three developers – now imagine this for a huge project with hundreds of developers from around the world. The situation can get unbearable. This is where the wonderful world of CVS comes in.

CVS is a system that will manage the code you and your developers work on and also managing how you and the developers interact and change the code. CVS will not only manage the code from all the developers, but will also attempt to resolve conflicts. CVS also has the ability to revert to previously written code right back to your first commits. CVS is in general a very flexible system, and numerous projects such as *KDE*, *GNOME*, *XFree86* et al. use it.

The CVS Structure

Before we get into the nitty gritty of using CVS, it is a good idea if we first look at how CVS actually works, and all the separate parts that are involved.

CVS is basically split into two parts – the client and the server. The client is the program that will enable the developer to get a local copy of the code from the server, and the client deals with sending code back to the server, adding files and directories and looking at log entries. The server is a program that deals with servicing the clients, and stores code in areas called Repositories. A repository is simply a directory on the server where the main copy of the code is stored.

Each developer will download a copy of the code and put it on their hard drive (this is called checking code out), and this local copy of the code is called the Working Directory. Now the code is in the Working Directory, the developer can edit the code where needed. When the developer has finished editing the code, the code can then be sent it back to the server – this is called Committing. Each commit the developer makes needs to have a log entry entered which is asked for by the CVS client.

Each time a developer commits some code back to the CVS, a new version will be applied to the file. An example is if we had a file called *myfile.cpp* and we initially check it in to the CVS, it will be given version 1.1. If we then edit the file and commit it, it will be then given a version of 1.2. Using the client and also some web front ends, we can revert back to any version number we wish. This is in many ways the true power of CVS in that you can always go back if you mess something up.

Getting started with the client

To demonstrate the use of the client, we are limited in some ways as I cannot give you a CVS server in which to test out checking in code. We will however look at checking out code and viewing logs by logging into an anonymous CVS server. Much of this part of the series will make sense when I show how to set up a CVS server in next month's installment, so keep this trusty copy of Linux Format around until then.

The first thing we need to do is to set up the client. This is pretty simple as most distributions have a cvs package available that you can install. On Debian you can also **apt-get install cvs**.

Checking out of cvs — the U indicates a module new to the current working directory.

```

cvs server: Updating kdeutils/kdf/doc/de
cvs server: Updating kdeutils/kdf/doc/en
cvs server: Updating kdeutils/kdf/pics
U kdeutils/kdf/pics/.cvsignore
U kdeutils/kdf/pics/Makefile.am
U kdeutils/kdf/pics/delete.png
U kdeutils/kdf/pics/hi16-app-kcmdf.png
U kdeutils/kdf/pics/hi16-app-kdf.png
U kdeutils/kdf/pics/hi16-app-kwikdisk.png
U kdeutils/kdf/pics/hi32-app-kcmdf.png
U kdeutils/kdf/pics/hi32-app-kdf.png
U kdeutils/kdf/pics/hi32-app-kwikdisk.png
U kdeutils/kdf/pics/mini-root.png
U kdeutils/kdf/pics/tick.png
cvs server: Updating kdeutils/kdf/unix_outputs
U kdeutils/kdf/unix_outputs/df-g.hpux
U kdeutils/kdf/unix_outputs/df-k.digital
U kdeutils/kdf/unix_outputs/df-k.hpux
U kdeutils/kdf/unix_outputs/df.man.hpux

```

Once you have got CVS installed, you can now set up the client. The client only needs to know one thing really – the CVSROOT of the CVS server. The CVSROOT is a line describing how to connect to the server. This looks something like this:

connection method : username@cvs.server.url : directory of the CVS repository on the server

Let's look at each of these:

Connection method This is the way in which the client connects to the server. There are basically three types:

- :pserver** – authorised password server where a password must be entered to access the server
- :local** – local access to a local repository on a hard disk
- :ext** – another type of access such as SSH

The kind of access used is up to the administrator of the CVS server. Ask the administrator if you are unsure.

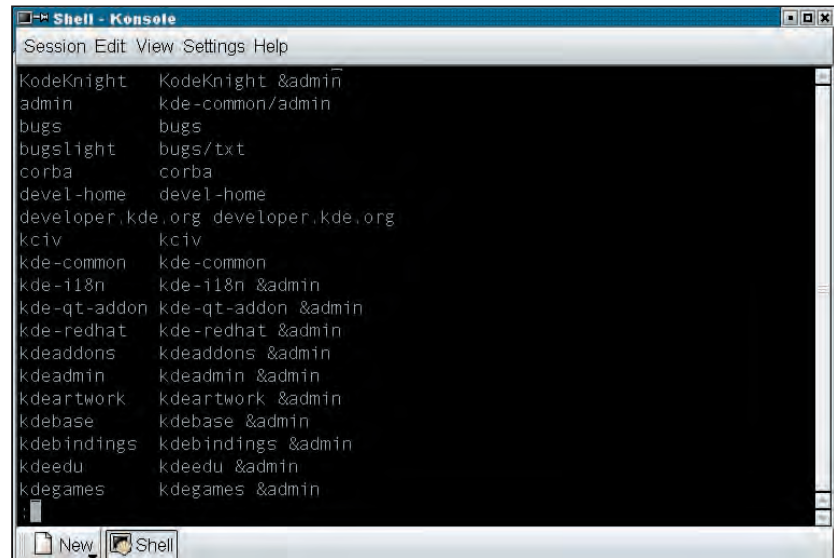
username@cvs.server.url This is the part which identifies your user account on the CVS server as well as the URL of where the server is located. The URL can be a domain name or an IP address.

Directory of the CVS repository on the server This is the absolute path of the location of the CVS repository on the server. An example of this would be something such as /cvs or /home/cvs.

To get us started, we will use the anonymous KDE CVS server as our server, and we will check out some code and look at some logs. The anonymous CVS won't allow us to commit code, but we can certainly play with checking it out. To start with the KDE CVS, we must set our CVSROOT environmental variable. We can set this in one of two ways: manually or configure the shell to set it for us. First of all, the CVSROOT for the KDE anonymous CVS is:

:pserver:anonymous@anoncvs.kde.org:/home/kde

So from our theory on the CVSROOT variable, we can see



that the CVS uses **:pserver** authentication (which requires a password), that the username is 'anonymous', the server is 'anoncvs.kde.org' and that the repository code is in /home/kde on the server. To set the CVSROOT:

Setting the CVSROOT for the current session Assuming you are using the *bash* shell, enter the following command:

export
CVSROOT=:pserver:anonymous@anoncvs.kde.org:/home/kde

Configuring the shell so it sets the CVSROOT for you To do this we need to edit *.bash_profile* which is in your home directory. This file is loaded whenever you login (which invokes a shell). You need to add the following to the file:

CVSROOT=:pserver:anonymous@anoncvs.kde.org:/home/kde
export CVSROOT

Remember to restart the shell or re-login to let the changes take effect. Once all this is set you can check that the variable has been set with:

echo \$CVSROOT

You should then see

:pserver:anonymous@anoncvs.kde.org:/home/kde shown as the variable. Now this is set we can get our feet wet and get something out of the server. First we need to login to the server. We can do this with:

cvs login

You will be prompted for a password – as this is anonymous CVS, we can just press enter as no password is set.

Seeing what is on the server

CVS stores things on the server in the way of modules. A module is like a directory on the CVS that contains a project or a series of related projects. A CVS administrator will update the list of modules available whenever new modules are added. We can take a peek at this list with the following command:

cvs co -c

This is broken down as follows:

- cvs** – We use the cvs command to issue an instruction to the CVS server.
- co** – This is shorthand for 'checkout'. The checkout (or co of course) is used to get modules from the CVS, and also to get information about the modules.
- c** – This is a switch to tweak to **co** command to show us the modules available on the CVS.

When we issue this command, we get something like this:



Installing CVS

Where to get it and what to do

Installing CVS is a fairly straightforward process, and is similar to the installation of other packages and source for Linux.

The first thing to do is to download the relevant packages. Many Linux distributions have prebuilt packages available which can be downloaded and installed using the package management tools in the distribution. Many distributions also provide the packages for download on their FTP sites. Check your own distribution for specific details.

Another site where you can gather CVS resources and downloads is <http://www.cvshome.org/>

When you have got the relevant packages for your system, you can install them using the following instructions:

Compiling the Source

If you got a source package, you need to first of all unzip and untar it into a directory:

tar xzf cvspackage.tgz

You can then read the **INSTALL** file on specific details of compiling the package, but you mainly need to:

./configure

make

make install

Installing an RPM

If you downloaded a prebuilt RPM package you can install it with:

rpm -i cvspackage.rpm

Installing from Debian

If you are using the Debian distribution you can install it with:

apt-get update

apt-get install cvs

LinuxFormatTutorialCvs

```

kciv      kciv
kde-common  kde-common
kde-i18n  kde-i18n &admin
kde-qt-addon  kde-qt-addon &admin
kde-redhat  kde-redhat &admin
kdeaddons  kdeaddons &admin
kdeadmin   kdeadmin &admin
kdeartwork  kdeartwork &admin
kdebase    kdebase &admin
kdebindings  kdebindings &admin
kdeedu     kdeedu &admin
kdegames   kdegames &admin
kdegraphics  kdegraphics &admin
kdeinstaller  kdeinstaller &admin
kdelibs    kdelibs &admin
kdemultimedia  kdemultimedia &admin
(abbreviated)

```

This listing gives us the name of the modules in the CVS. If you are wondering why each line contains the same module name twice, this is because it refers to the path on the CVS (in /home/kde) of where the module is stored. Typically this is the same name as the module and is hence the same.

Checking out a module

Before we checkout a module, we need to find a nice place to keep our sources. I usually create a directory /sources where I keep mine, and then have a subdirectory for each CVS server. So for my KDE sources I have them in /sources/kde. You can of course put them anywhere (if you have the permissions) but /sources is a good start.

To get a module from the CVS is pretty simple. Let's first choose a module that we want to look at - *kdegames* for example. We can check it out with:

```
cvs co kdegames
```

This will checkout the module into the current directory. When you issue the command, you will see something like:

```

cvs server: Updating kdegames
U kdegames/.cvsignore
U kdegames/AUTHORS
U kdegames/COPYING
U kdegames/COPYING.LIB
U kdegames/ChangeLog
U kdegames/INSTALL
U kdegames/Makefile.am
U kdegames/Makefile.cvs
U kdegames/README
U kdegames/config.h.bot
(abbreviated)

```

As you can see from the output of the command, each file in the module is checked out, and beside it is a letter U. This letter is a code for the status of the checkout or update; the letter means the file is new to the working directory. Another common letter you will see is the letter P (particularly when you update your sources). The letter P means that the file has been modified and the changes are being (P)atched into the file.

Updating your source code

It has been a couple of days since I checked out my kdegames module, and I would like to see if there are any changes and updates to the source code. I can check this by updating the module and changes will be merged into my existing code. The way this works is through some patch/diff magic. Only the updated code is added to the file and the file is not re-checked

cvs does its stuff — there's a lot to check out of KDE and this is only the beginning of the alphabet.

out — this is good as you can update large modules efficiently on a modem connection as the bandwidth requirements for the changes is minimal.

To get the wheels in motion we will use the **cvs update** command to perform the update. To do this, go to the directory where the module you want to update is stored (in my case /sources/kde/kdegames) and enter:

```
cvs update -PA
```

Let's first take a look at those command line options I added:

- P** — This prunes the working directory (on your disk) of any directories that were removed on the server.
- A** — This resets any special tags that have been applied to the file. This always ensures you get the main development version of the file.
- d** — This checks out any directories that are on the server that may not be in your working directory.

When this command is entered, I start getting a listing of the files in the module and what cvs has done with them (indicated with the letter on the left like with the 'cvs checkout').

When I updated *kdegames* here is an example of the updates had:

```

P doc/kmines/index.docbook
U doc/kmines/kmines1.png
U doc/kmines/kmines2.png
cvs server: Updating doc/konquest
P doc/konquest/.cvsignore
P doc/konquest/Makefile.am
U doc/konquest/index.docbook
cvs server: Updating doc/kpat
U doc/kpat/.cvsignore
P doc/kpat/Makefile.am
U doc/kpat/clubs.png
U doc/kpat/hearts.png
(abbreviated)

```

Here we can see for example that index.docbook was modified by a developer; hence it has a 'P' on the left of the line. We can also see that *kmines1.png* was a new addition to the local copy (shown with the 'U'). This system of showing what was updated and in which way shows you clearly how many updates there were since your last checkout or update.

Viewing logs

So far we have covered most things in terms of getting and updating modules from the CVS server. Let us assume now that

GUI CVS Clients

Although CVS is a command line tool, there are also some unofficial graphical clients which act as a front end to the CVS client.

Cervisia - <http://cervisia.sourceforge.net/>

Cervisia is a KDE interface to CVS. The application will perform most operations of the command line client and there is also a *Konqueror* plugin for viewing CVS repositories. The plugin let's you commit files contextually.

LinCVS -

<http://www.lincvs.org/>
LinCVS is a Qt based frontend. *LinCVS* is quite a capable client and similar to *WinCVS* on Windows in it's design. Again, most facilities in the command line client work with *LinCVS*.

TkCVS - <http://www.twobarleycorns.net/tkcv.html>

TkCVS is a client written Tcl/Tk and hence is portable across Tcl/Tk supported platforms. *TkCVS* works quite well and the module browser is nice. A good application all round.

someone has modified a file, and the changes have been merged into your local copy. What is difficult for us to determine is exactly what changes the developer made to the file in terms of functionality.

It is this problem that CVS logs are designed to solve. Each time you commit a file to the CVS server you will be asked for a log entry to summarise what changes you made, and we can use our handy cvs client to poke at these log entries. Another side of the log facility is that it can be bizarrely fascinating just looking at log entries on anonymous CVS servers like the KDE one we have been using.

Let's take a look at some log entries for a file in the *kdegames* module. We will use as a test case the *kdegames/kbackgammon/kgbstatus.cpp* file. Just go to that directory and type:

```
cvs log kbgstatus.cpp
```

We will then get some log entries for the file. The last two updates are shown here for brevity:

```
-----
revision 1.5
date: 2001/05/02 04:59:59; author: hoefkens; state: Exp;
lines: +8 -25
Minor cleanup. Removed dynamic allocation of board_ array.
-----
revision 1.4
date: 2001/05/02 04:41:09; author: hoefkens; state: Exp;
lines: +9 -6
Removed the upper limit of 64 for the cube. Unlike FIBS,
GNUbg does not limit the max. value of the cube. However,
we still enforce the cube to be a power of 2.
-----
```

If we take a look at each entry we can see the date the file was updated, the author (hoefkens is the name of the CVS account the author has) and the log entry that the author added to summarise the changes.

Committing to the CVS

All of this is interesting for using a CVS server as a read only service, but at some point you may want to contribute some code to the server. To do this you will first need to ask the CVS administrator for an account (if one is required), and then you can then make whatever modifications you need and merge your updates back into the CVS. To demonstrate this facility I will issue commands on my own CVS server.

I currently have a fully up to date checkout of the *kwebstat* module from the CVS server. I will now make a trivial change to the *kwebstat/README* and *kwebstat/kwebstat/kwebstat.cpp* files. With these changes, I can commit them to the server. All I need to do is go to the toplevel directory in the *kwebstat* module and issue:

```
cvs commit
```

CVS will then analyse all of the files in the *kwebstat* module and then prompt me for a log entry. You will then see something like this in your default editor:

```
CVS: -----
CVS: Enter Log. Lines beginning with `CVS:' are removed
automatically
CVS:
CVS: Committing in .
CVS:
CVS: Modified Files:
```

```
CVS: README kwebstat/kwebstat.cpp
```

```
CVS: -----
```

At this point the log entry is added above the lines (the lines are removed by CVS). Note how the files which have been modified by CVS are shown - it is always a good idea to check that the right files are being committed.

Once a brief log entry has been added, save the file and you will then see the files committed and new version numbers added:

```
Checking in README;
/cvs/kwebstat/README,v <-- README
new revision: 1.3; previous revision: 1.2
done
*
Checking in kwebstat/kwebstat.cpp;
/cvs/kwebstat/kwebstat/kwebstat.cpp,v <-- kwebstat.cpp
new revision: 1.5; previous revision: 1.4
done
```

Adding and removing files

Aside from committing changes to files, you may need to add files. This can be done with:

```
cvs add thefile
```


Replace thefile with the file you wish to add. You will need to add the file to the working copy of the module before you can use **cvs add** to add it, and the file will be added to the same directory on the server. Once you issue the **cvs add** command, you will need to also issue a **cvs commit** to add it. The reason for this is that **cvs add** only schedules the file for addition in case you need to commit other files also (such as modified files).

You can also remove a file by first removing it from the directory in the working copy and then issuing:

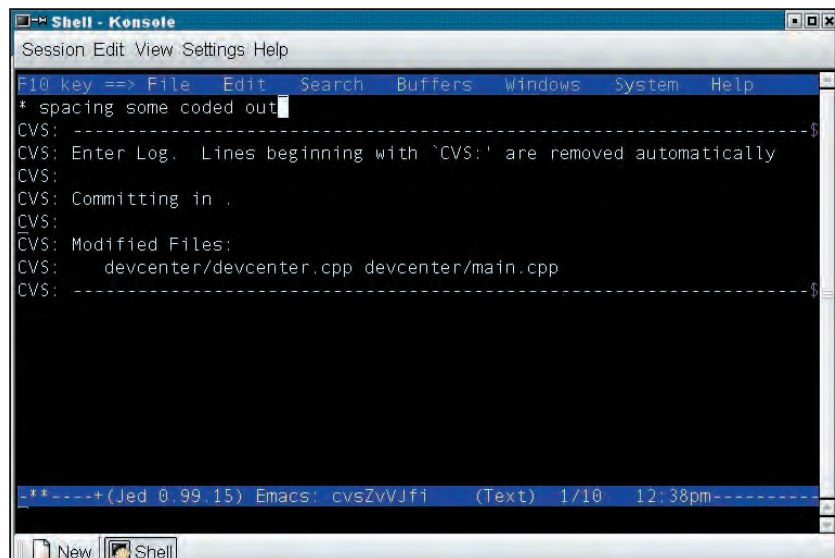
```
cvs remove thefile
```

You will need to **cvs commit** for this also.

Conclusion

CVS is a powerful and flexible tool that can greatly simplify development and minimise conflicts. In this issue we have looked at the main features of the CVS client. We will really put all of this information to ultimate use next month when I will show you how to set up a CVS server. We will cover setting up the server, creating repositories, adding branches and also setting up *cvsweb* to create a web based interface. Stay tuned... 

cvs commit prompts you for a log entry.



INTERPROCESS COMMUNICATIONS

Writing client/server programs



This month **Charlie Stross** introduces signals, then gets out his plumbing gear and plugs into sockets and pipes.

Linux is a POSIX-compatible UNIX like operating system. As such, it has a whole grab-bag of features for letting one program talk to another. In the beginning, UNIX System 7 provided pipes, signals and temporary files. The BSD versions added TCP/IP networking and sockets, while AT&T's UNIX team added a slew of IPC (inter-process communication) options, including shared memory, semaphores, and streams (not really used on Linux). Then the Bell Labs Plan Nine research team, who got heavily into doing esoteric things with filesystems, invented the `/proc` system and the Linux kernel developers snaffled it and added their own extras, notably the shared memory filesystem (which makes shared memory easier to use).

The result is that Linux is a rich environment for writing client/server software, although not all the mechanisms you'll find there are portable to other operating systems. A SHM-filesystem based solution won't run on Solaris or other SVR4 UNIXes, and SVR3/SVR4 shared memory won't work on BSD. So I'm going to keep to the limited subset of client/server tools that are relatively portable — signals, pipes, and sockets.

In this tutorial, I'm going to go over some of the basics of interprocess communication — signals and pipes, with a side-order of sockets. In next month's tutorial we'll see how we go about building socket-based internet clients and server programs (such as a simple web server).

Signals

Signals are about the most primitive way for two processes (executing programs) to communicate: they don't transfer any information at all, except the fact that a signal with a given number has been raised — because some condition has occurred. For example, if a process divides by zero, SIGFPE (floating point exception) is sent to the process. A lot of other conditions can trigger signals; for example, hitting the interrupt key on a TTY that a process is connected to sends a SIGINT (interrupt) signal, and the **kill** command can be used to send arbitrary signals. You can see a list of the signals supported by Linux in the man page `signal(7)`; it's a superset of POSIX.1 and SUS v2 (standard UNIX specification).

Processes can do three things: ignore the signal, allow a default action to occur (e.g. terminate the process — this is the default for SIGFPE), or provide a function that is called when the signal occurs. Perl processes can set handler functions, and they can also send signals. Here's how it works.

There's only one signal that doesn't affect the target process:

signal 0. This signal merely checks whether the process is alive and hasn't changed its user ID; if **kill(0, \$some_process_id)** fails, then **\$some_process_id** isn't accepting signals. (Maybe it's a zombie?)

To send a signal in Perl, we use the **kill()** command. **kill()** takes two or more parameters — the signal to send, and a list of process ID numbers. **kill()** returns the number of processes successfully signalled. Signals can be represented in numeric form (e.g. 9 for SIGKILL) or symbolic form (e.g. HUP for SIGHUP). The list of process ID's are just the PIDs of the processes to signal. Note that your Perl script can't kill processes belonging to higher-privileged user ID's — if you want to kill one of root's processes, you have to be root, and if you do this from a CGI script running with a UID of 'nobody' or 'httpd' you can only signal other processes owned by 'nobody' or 'httpd'.

Secondly, there are two ways of processing a received signal in Perl: the traditional way, and the **sigtrap** pragma.

Perl (this is the traditional way) provides a hash called %SIG, in the main namespace. You can create subroutines to handle specific signals, and store them in %SIG; for example, if you write a subroutine called **handle_hup()** to handle SIGHUP (hang-up) signals, you'd put a reference to it (or just its name) in

\$SIG{HUP}:

```
$SIG{HUP} = \&handle_hup();
```

Thereafter, whenever the process receives a SIGHUP, the subroutine **handle_hup()** is triggered.

You can also assign the name of a subroutine to %SIG:

```
$SIG{HUP} = 'handle_hup';
```

Or an anonymous subroutine:

```
$SIG{HUP} = sub {die "Killed by SIGHUP!\n"};
```

Now, here's a problem: signals are asynchronous. They can come at you from anywhere and at any time. To handle asynchronous events like this, you need to write re-entrant code — code that can be invoked from within itself without losing track of where the program counter and registers has gotten to. This isn't re-entrancy at the Perl level — it's at the C or assembly language level, the level Perl is written in. Perl is linked to *libc*, and older versions of *libc* are not re-entrant — *GNU Libc*, aka *Libc 6*, is, but you can't count on older versions of Linux providing this.

This means that your Perl signal handler may go bang messily if it receives a second signal while trying to handle a signal. So you need to keep signal handler subroutines short and sweet. Ideally all they should do is log an error message somewhere, close any open system resources (such as open filehandles), and call **die()** gracefully. Even doing that much can be a bit risky.

Note that the %SIG array is global by default; if you want to change a signal handler locally, say within a subroutine, you should override %SIG using **local()**, so that when you leave the special scope %SIG regains its normal settings.

The shiny new Perl approach to signals is the **sigtrap** pragma (a pragma is a built-in feature controlled using **use()** that changes the way the compiler behaves). Sigtrap lets you specify that one of a number of predefined behaviours is to be triggered by one or more signals. Behaviours include 'stack-trace' (output a Perl stack trace and dump core), 'die' (die with a programmer-specified message), and 'handler', which is followed by the name of a user-written subroutine to use (as in %SIG). Lists of signals to apply a behaviour to include 'normal-signals' (HUP, INT, PIPE, or TERM), error-signals (ABRT, BUS, EMT, FPE, ILL, QUIT, SEGV, SYS and TRAP — these would usually cause your Perl interpreter to die), 'old-interface-signals', 'untrapped' (all signals for which no handler has yet been installed), and 'any' (all signals, period). You can also list signals by name or number.

For example:

```
use sigtrap qw(stack-trace normal-signals);
```

Causes Perl to install the stack-trace handler for all the signals listed in the 'normal-signals' set.

```
use sigtrap qw(die untrapped);
```

Makes the **die()** handler the default for any signals we haven't previously set up a handler for: and:

```
my $my_handler = sub { warn "SIGHUP received\n" };
use sigtrap 'handler', $my_handler(), HUP;
```

Installs a handler for SIGHUP that prints 'SIGHUP received' on STDERR.

101 Uses for a dead signal

What can you usefully do with **kill()** and %SIG?

One useful trick is SIGALRM, the alarm signal. When you call the Perl command **alarm()**, it sets a timer; after the specified number of seconds, this causes the kernel to send a SIGALRM signal to the current process. (You can only have one timer active at once, each time you call **alarm()** it deletes the previous timer; you can also manually cancel an alarm by calling **alarm(0)**. **alarm()** returns the amount of time remaining on the previous timer.)

Suppose you want to read some data from another process that's writing to a pipe — but which might die of boredom or something. If you just read like this:

```
while ($input_data eq "") {
    $input_data = <MY_PIPE>;
}
```

you risk being blocked forever by whatever process is on the other end of the file handle MY_PIPE.

You can avoid this problem by using SIGALRM and a custom signal handler:

```
eval {
    my $alarm_timeout = 60; # wait up to sixty seconds
    before giving up
    local %SIG;
    $SIG{ALRM} = sub { die "Error: MY_PIPE coprocess timed
    out\n"; };
    alarm $alarm_timeout;
    while ($input_data = <MY_PIPE>;) {
        $input_data = <MY_PIPE>;
    }
    alarm 0;
}; # end of eval block scope
alarm 0;
if ($?) { print "Error: $@\n"; }
```

If the alarm goes off inside the **eval()** block, the appropriate action to take is to call **die()** — otherwise Perl might try to restart the **syscall** to read from MY_PIPE. By calling **die()** in an eval

block we raise an exception and exit the block. If the alarm goes off, the result will be that \$@ holds the string "Error: MY_PIPE coprocess timed out".

(There's a slightly more convoluted example in Chapter 16 of *Programming Perl*, 3rd edition — involving nested **eval()** scopes to get around a problem with a **syscall** that may not be implemented on some platforms. By ignoring cross-platform portability and focussing on Linux, we get off lightly!)

Here's another use for a signal (which we'll see an example of later): signalling a process group. On Linux, processes have a family tree. When you want to run a new program (a running instance of a program is called a process), your current program must use the **fork()** system call. **fork()** clones the current process, and returns the process ID of the child to the parent; the only distinguishing features of the new child are its new process ID and a couple of bits of meta-information used by the kernel. If you want to run a different program, the child must check its process ID and call **exec()**, to execute the new code (overwriting itself in the machine's memory, roughly speaking). Ultimately all processes are descended from *init*, the root process.

We refer to a parent and its child processes as a 'process group'; process groups have a number corresponding to the process ID of the process group leader (or parent). Perl processes can check their own process ID, using the special variable \$\$, and collect the PIDs of children as they call **fork()**. You can individually signal a whole bunch of children:

```
kill(HUP, @my_kids);
```

But if they've spawned grandchildren or other descendants, you may not have a list of their PIDs. The solution is to become a process group leader, set a signal handler for signal 1, SIGHUP, and then send a HUP signal to a negative number corresponding to the process ID of the process group leader. (Send a signal to, say, 407, and it affects process ID 407. Send it to -407 and it goes to both process 407 and all its descendants. So, if you send a signal to -\$\$, it goes to all the processes descended from \$\$.) For example:

```
eval {
    setpgid(0, $$); # set $$ to be the process group ID
};
if ($?) {
    die "Could not setpgid(0, $$): $@\n";
}

# do stuff that spawns child processes here,
# including shell scripts or daemons
:
# now it's time to shut down all descendants gracefully
```



Pipe pointers

Important note: pipes connect two processes and they are unidirectional. That is: a given Perl program can read from one end of a pipe, or can write to one end of a pipe, but can't read and write to the same end of the same pipe. If you need bidirectional communication between two processes (call them A and B) you will need to use a pair of pipes, one which A writes to and B reads from, and one which B writes to and A reads from.

A second characteristic of pipes is that they're buffers. You can pump data into a pipe,

but if the process at the other end doesn't suck data out fast enough it can block you. (Clue: this is a good reason to practice using SIGALRM to time out operations that rely on another process.)

Finally, there are two different types of pipe on Linux — anonymous pipes (the normal kind) and named pipes (which have names in the filesystem, and which you can open and close like a file). You also want to avoid getting pipes confused with sockets (which we go into at great length later on.)

LinuxFormatTutorialPerl

```
local$SIG{HUP} = 'IGNORE'; # we don't want to kill ourselves
kill(HUP, $$); # signal entire process group
```

When processes exit, the parent is expected to tidy up after them. A signal – SIGCHLD – gets sent to the parent, which must then use one of the **wait()** or **waitpid()** system calls to acknowledge the death in the family. Processes that don't get **wait()**'d for hang around as entries in the process table, and are known as zombies: when writing a program that spawns lots of children you need to take care of this. SIGCHLD was intended to allow parents to do things like wait for a child to run and terminate before continuing. Perl's **system()** and backquote mechanisms automatically take care of reaping CHLD signals, but if you call **fork()** directly to spawn children, you need to be able to deal with the consequences manually.

The simplest approach to dealing with a death in the family is to callously ignore it:

```
$SIG{CHLD} = 'IGNORE';
```

Alternatively, you can subcontract the job out to a grim reaper — a subroutine that periodically checks for zombies, and calls **waitpid()** to collect them. This is the recommended tactic for server processes (like the ones we'll eventually get round to writing).

For example (by way of the Camel book):

```
our $zombies = 0; # global counter for zombies
$SIG{CHLD} = sub {$zombies++};
```

```
sub reaper {
    my $zombie;
    our %exit_status;
    $zombies = 0;
    while (( $zombie = waitpid(-1, WNOHANG) ) != -1) {
        $exit_status{$zombie} = $?;
    }
}
```

```
# now we do a whole lot more stuff until finally ...
```

```
while (1) { # main execution loop for the server
    if ($zombies > 0) {
        reaper();
    }
    # and so on
}
```

The idea of this approach is that whenever a child dies, the global counter `$zombies` is incremented. Servers tend to run in a continuous loop; if `$zombies` shows that there are pending children waiting to be reaped, the main loop periodically kicks off the **reaper()** subroutine.

The **reaper()** simply runs **waitpid()** repeatedly, recording the returned child process ID's (returned in `$zombie`), until **waitpid()** returns -1, indicating that there are no dead child processes.

Pipes

A pipe is a one-way i/o channel that funnels bytes from one process to another. Unlike signals, which merely convey a single bit of information (such as "hello, process 4117! Your child process 4118 has died!"), pipes let us transfer arbitrary data from one process to another. They're not the only way of doing that – temporary files and shared memory segments can be used for the same job – but pipes are often very useful because they're easy to use in shell programming, and being able to access them from Perl lets us write Perl scripts that can interact with pipelines.

In Perl, there are two general strategies for using pipes. You can use a pipe to talk to another program by calling **open()** with some special arguments; or you can use the low-level **pipe()** command to create a couple of pipes, and then **fork()** so that a parent process can talk to its children.

Talking to aliens

We're used to seeing Perl programs open files for reading data or writing results like this:

```
open(INPUT, "</var/spool/maillog") or die "Couldn't open
maillog: $!\n";
```

However, Perl's **open()** command can also be used to run an external program and connect a pipe to it — *either* to write data from your Perl program to the external command, *or* to let your Perl program read data from the external program via a pipeline.

Examples:

```
open(OUTGOING, "| sendmail -ba") or die "Can't send mail:
$!\n";
```

Opens a pipe into a copy of sendmail, presumably so that the Perl program can send an email message.

```
open(MONITOR, "netstat -an 2>/dev/null |") or die "Can't
fork: $!\n";
```

Runs **netstat -an**, discards the standard error, and attaches it to the filehandle MONITOR, so that we can do something like this:

```
while ($data = <MONITOR>) {
    # do something with output from netstat
}
```

When you use **open()** this way, Perl implicitly calls **fork()** and **exec()** to spawn a sub-process. This is not necessarily the program you expected to run: if the argument you give to **open()** has more than one token in it, **open()** spawns a shell and passes your command line to the shell, while connecting the standard output from the shell to your filehandle (if you're trying to read from a pipe), or your filehandle to the standard input of the shell (if you're trying to write to it). So it's legal to do something like this:

```
open(WORDS, "find . -type f -name '*.txt' -exec cat {} \; | wc
-w |");
```

The output from the pipeline (find searching for files and calls cat on them; wc then counts the number of words output by cat) is then available for reading on the Perl filehandle WORDS. However the pipeline command isn't executed directly by Perl, but by `/bin/sh`.

Talking to yourself

Using **open()** to read from or write to external processes is all very well, but what if you want to read and write to the same process? Or if you want to spawn a child copy of your Perl process to carry out some task, then read some data back in from it?

The traditional UNIXy way of doing this is pretty low-level. First, you use the **pipe()** system call to create a pipe. This returns two filehandles — a reader and a writer. If you want bidirectional communication with your child, you call it twice, giving a total of four file handles (one for the parent to write to, which has a corresponding one for the child to read from, and one for the child to write to, with a corresponding reader handle for the parent). You then call **fork()** to spin off a child. The child now has to close the parent filehandles, while the parent closes the child handles. Then you have to somehow manage i/o without a deadlock occurring.

Deadlock occurs when two processes are both waiting for

each other to do something, and it's deadly. If you hit a situation where the parent is waiting for the child to say something, but the child expects the parent to say something, both processes will hang — unless you've set up a SIGALRM handler and set an alarm to go off and break one or other of the processes out of the deadly embrace.

You might think you can avoid this situation by carefully planning your parent and child dialog, but there's a fly in the ointment: UNIX I/O buffering. Normally, input and output on filehandles are buffered — that is, the standard I/O library saves up bytes of input until it has enough to fill a buffer which can be transferred to a reader process in a lump. This is an important optimization for UNIX systems — otherwise the kernel would have to switch between executing the writer and the reader processes for every byte transferred, and context switches are expensive — but if you just write a short string, and expect a response, you may be disappointed, because the short string won't fill the I/O buffer and therefore the client won't get to read it.

A partial solution to this is to tell Perl to use unbuffered I/O on a filehandle. You can do this using select:

```
select FH;
$| = 0;
```

The `$|` special variable controls I/O buffering; setting it to 0 disables buffering on the current selected filehandle only, so if you have a bunch of filehandles in use you may need to select and unbuffer all of them in turn.

The standard Perl distribution comes with a couple of lifebelts for programmers who really want to write processes that do this sort of thing. First, there's the module `IPC::Open2`. `IPC::Open2` supplies a single subroutine, `open2()`; this does what you'd intuitively expect of the (illegal) command:

```
$pid = open(HANDLE, "| some_pipeline |");
```

— that is, `open2()` takes two filehandles as arguments, and glues them onto the standard input and standard output of a command. For example:

```
open2(\*INPUT, \*OUTPUT, $my_external_filter);
```

You can now write to `INPUT` and read from `OUTPUT`. If `open2()` succeeds it returns the PID of the child program; if it fails it raises an exception that you can trap with an eval block:

```
eval {
    open2(\*INPUT, \*OUTPUT, $my_external_filter);
};
if ($@ =~ /^open2:/) {
    warn "failed to spawn $my_external_filter: $@";
}
```

Note that `open2()` doesn't reap children, so you'll need to trap `SIGCHLD` (as described earlier) in programs that call it.

Sockets

Signals and pipes have a major deficiency when we use them for interprocess communication; they're restricted to a single computer. If you want to do network programming, or write general purpose client/server applications that can run on more than one machine, you need to look into sockets.

The easiest way to think of a socket is as a bidirectional pipe (yes! you can read from and write to the same socket) that, instead of connecting to the standard input and standard output of a process on the same computer, connects to a process bound to a TCP/IP address and port number on a computer. The biggest 'gotcha' is to remember that while the socket itself is bidirectional, the work of setting up the connection requires distinctive and different tasks to be carried out by a server and

its respective client application.

To set up a socket-based connection, a server must first create a socket, tell it what IP address and port number to bind to, and call `listen()` to establish a queue of incoming connections. The server also needs to know what to do when an incoming connection arrives.

The client's job is a bit easier: it needs to create a socket, tell it the address of the remote machine, and then call `connect()` to talk to the server.

There are several ways to write both a server or a client in Perl: suffice to say that the easy way is to use the standard `IO::Socket::INET` class, which provides a socket object. (Or you can use the lower-level `Socket.pm` module to provide low-level functions, for example if you want to mess around with the protocol type or set various flags on the socket.)

For now, here's a very simple client that talks to a web server and requests the default page:

```
use IO::Socket::INET;

my $remote = "www.linuxformat.co.uk"; # a well-known
web server ;-)
my $http = 80;                        # standard port for
HTTP traffic

my $sock = IO::Socket::INET->new(PeerAddr => $remote,
                                PeerPort => $http,
                                Proto => 'tcp',
                                Type => SOCK_STREAM)
    or die "Couldn't connect to $remote:$http: $!";

print $sock "GET /\n\n"; # a minimalist HTTP request

print STDOUT $sock->getlines();
close $sock;
```

Note the line `print STDOUT $sock->getlines()`.

`IO::Socket::INET` inherits a bunch of methods from `IO::Handle`, a very neat abstraction that lets us treat file handles as objects. Sockets are treated much like filehandles in Perl, and `getlines()` simply reads an array of lines from the filehandle. All the real socket-handling voodoo is concealed in `IO::Socket::INET::new()`, which accepts a bunch of parameters but defaults to sensible values when setting up a new socket connection.


Here's a really simple server. It listens for connections to the local machine on port 999, and when one arrives, prints the current date and time to the client:

```
use IO::Socket::INET;

my $port = 9999; # some unused TCP port to bind to

$server = IO::Socket::INET->new(LocalPort => $port,
                                Type => SOCK_STREAM,
                                Reuse => 1,
                                Listen => 10)
    or die "Couldn't bind to port $port: $!";

while ($client = $server->accept()) {
    $client->print( scalar(localtime(time)), "\n");
}
```

Warning: don't try and turn this into a general purpose server! For a whole bunch of reasons (which we'll look at in depth next month), this server won't scale up. 

High level solutions

If all you want to do is talk to a specific type of internet server — that is, a program on a remote machine that uses a specific protocol, such as **SMTP** for mail routing or **HTTP** for serving up web content — there are very-high level modules (**Net::SMTP** and **HTTP::Request**) that will let you issue requests without having to deal with sockets at all. But socket programming is unavoidable if you want to implement your own special services.

NEXT MONTH

Next month we're going to see how to do it properly — how to write a server that will cope with a whole bunch of clients and do useful work.

REFLECTION

Speaking Java



Java programs are self-aware - as **Richard Drummond** explains with this introduction to Java's reflection API.

A variable, in object-oriented programming languages such as Java, may refer to run-time objects of different classes; this is one of the ways that the core object-oriented principle of polymorphism is implemented. A consequence of this parametric polymorphism is that the class of an object referred to in a program may be unknown until run-time. In C++, the class is really just a construct of the compiler, so nothing is known about classes at run-time. However, in Java, all classes are represented as objects which are accessible to and can be queried by the running program. You can, for example, examine a class and find out what methods it supports and the number and type of the parameters each method accepts.

All classes in a running Java application are represented by instances of the class **Class**. Every array is also represented by a class object shared by all other arrays of the same element type and number of dimensions, and the primitive types boolean, byte, char and so on and the empty type void are also all represented as class objects. The class **Class** provides methods to find the name and superclass of a **Class** object and to query what constructors it supplies, what methods it declares and what interfaces it implements. **Class** itself has no public constructor: class objects are instantiated by the JVM as classes are loaded.

This ability for a Java program to interrogate its own structure is known as reflection – an apt name for a mechanism which lets a program hold up a mirror and examine itself. The reflection API has been available since *JDK 1.1* and is implemented by the

package **java.lang.reflect**, building upon on the basic properties of the Java classes **Object** and **Class** that we have just discussed.

Reflection has many applications, but a principal use is in automated development tools. Java's component architecture, JavaBeans, relies heavily on reflection, and it is reflection that allows rapid development tools such as Borland's *JBuilder* query the methods and properties of a Bean and let you manipulate them with a graphical interface. More on Java Beans in a later article, however.

Knowing your class

To make use of reflection, you first have to find the class of the run-time object in question. There are several way of doing this, the most general of which is simply to invoke the method **getClass()** on the object. **getClass()** is defined by the root class **Object** and so is inherited by all Java classes.

```
Class c = some_object.getClass();
```

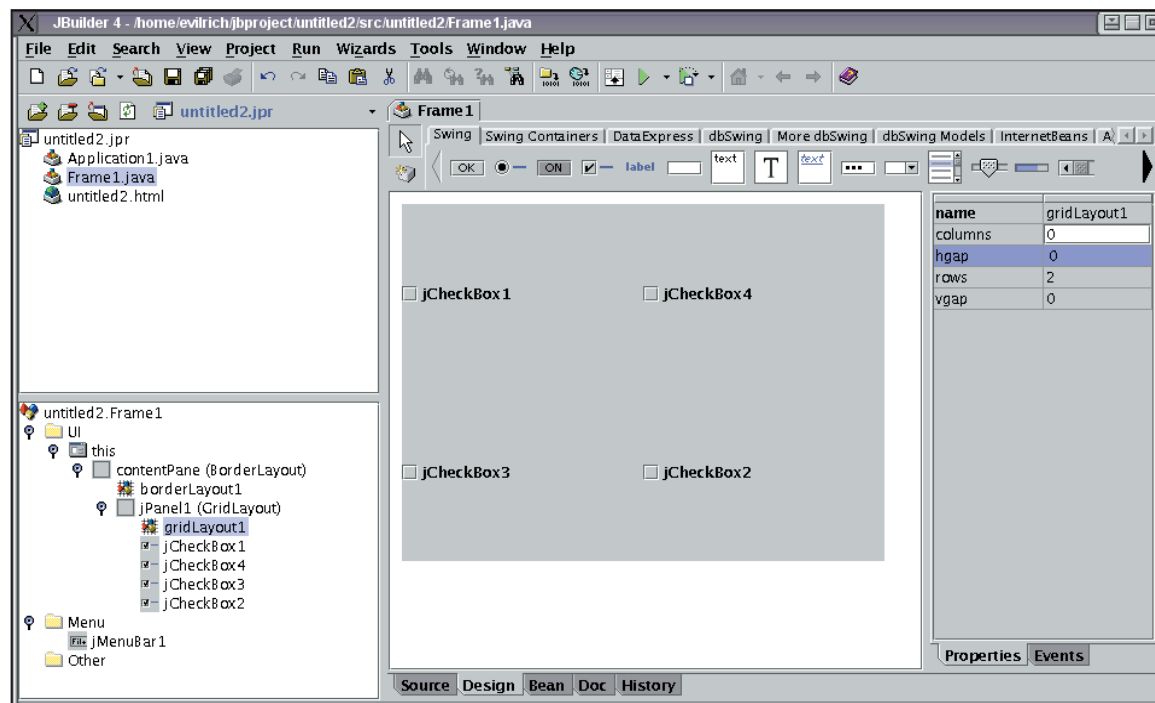
Alternatively, if you know the class at compile time, you can use this method:

```
Class c = java.awt.Frame.class;
```

Every class appears to have a publicly accessible field called **class** which is a reference to its **Class** object, but really this is just syntactic sugar for a purely compile-time construct.

The third approach is to use the static method **forName()** from the class **Class** and is useful if you know the name of the class at run-time but not at compile-time (perhaps the class name is being supplied by the user or is read from a

Visual development tools such as *JBuilder* rely on Java's reflection mechanism.



configuration file):

```
Class c = Class.forName( class_name );
```

Here **class_name** is a **String** containing the fully-qualified name of the class, for example, **java.awt.Button**. If the specified class has not already been loaded, the method **forName()** loads and initializes it. It can throw an exception of the type

ClassNotFoundException if the class cannot be located and this exception must be caught or declared to be thrown.

Once you know an object's class you can start interrogating and manipulating it. You can invoke **getMethods()** to find a list of the public methods it supports or you can call **newInstance()** to instantiate a new object from that class.

Have a look at this example:

```
import java.lang.reflect.*;
public class DumpMethods
{
    public static void main( String args[] )
    {
        try {
            Class c = Class.forName( args[0] );
            Method m[] = c.getDeclaredMethods();
            for( int i = 0; i < m.length; i++ )
                System.out.println( m[i].toString() );
        }
        catch( Throwable e ) {
            System.err.println( e );
        }
    }
}
```

DumpMethods accepts a fully-qualified class name as an argument, loads that class and prints out all the methods it declares (both public and private and those inherited from superclasses). The class **Method** represents – surprise, surprise – a class method and is part of the package **java.lang.reflect**. Calling the **Method** object's **toString()** causes it to return a string containing the method specification formatted in a familiar way, ready for us to print out.

Try running **DumpMethods**, remembering to supply the full name of the class you want examine. For example:

```
java DumpMethods java.awt.image.BufferedImage
```

A concrete example

If you think, at this point, that reflection is all very clever, but not at all practical, then have a look at the next example. This is an extract from class which represents a JDBC connection to a database server. The constructor opens the connection and takes the name of the JDBC driver to use and the a URL pointing to the database to open as arguments.

```
import java.sql.*;

class myDatabase
{
    private Connection connection;


    public myDatabase( String driver_class, String url )
    {
        try {
            Class.forName( driver_class ).newInstance();
            connection = DriverManager.getConnection( url );
        }
        catch( ClassNotFoundException e ) {
            System.out.println( "Sorry, I could not find the database
```

```
driver "+driver_class+" " );
        }
        catch( SQLException e ) {
            System.out.println( "Sorry, I could not open the database
            "+url+" " );
        }
        catch( Exception e){
            System.out.println( "There was a problem using the
            database driver "+driver_class+" " );
            System.out.println( "with the database "+url+" " );
            System.out.println( e.getMessage() );
        }
    }
}
```

Here's some code which you could use to test the above class:

```
class Test
{
    public static void main( String args[] )
    {
        myDatabase db;
        db = new myDatabase( "org.gjt.mm.mysql.Driver",
            "jdbc:mysql://localhost/text?user=rich&password=squiggle" );
    }
}
```

The first task of the **myDatabase** class's constructor is to ensure that the JDBC driver for the database is loaded. It does this by calling **Class.forName()** with the class name of the driver. This gets loaded and initialized and hence the driver registered with JDBC. It can then be used to in the following statement to open a connection to the specified database.

Now, this is practical example of reflection. We don't want to hard-code the class name of the database driver into an application. Rather we would specify it in a configuration file, so that we can easily change to use a different database engine without requiring the code be modified and re-compiled. 

Method Summary

How to do it with class

Here is an overview of some of the classes and methods that are useful when using reflection:

class: **java.lang.Object**

methods:

public final Class getClass()

Returns the run-time class of an object as an object of type **Class**.

class: **java.lang.Class**

methods:

public static Class forName(String className) throws ClassNotFoundException

Attempt to load the specified class or interface and return its associated class object as an object of type **Class**.

public Object newInstance() throws InstantiationException, IllegalAccessException

Creates a new instance of the class represented by this **Class** object.

public boolean isInstance(Object obj)

Determines if the specified **Object** is assignment-compatible with the object represented by this **Class**.

public String getName()

Returns the fully-qualified name of the entity (class, interface, array class, primitive type, or void) represented by this **Class**, as a **String**.

public Class getSuperclass()

Returns the **Class** representing the superclass of the entity (class, interface, primitive type or void) represented by this **Class**.

public Method[] getMethods() throws SecurityException

Returns an array containing **Method** objects reflecting all the public member methods of the class or interface represented by this **Class** object, including those declared by the class or interface and those inherited from superclasses and superinterfaces.

class: **java.lang.reflect.Method**

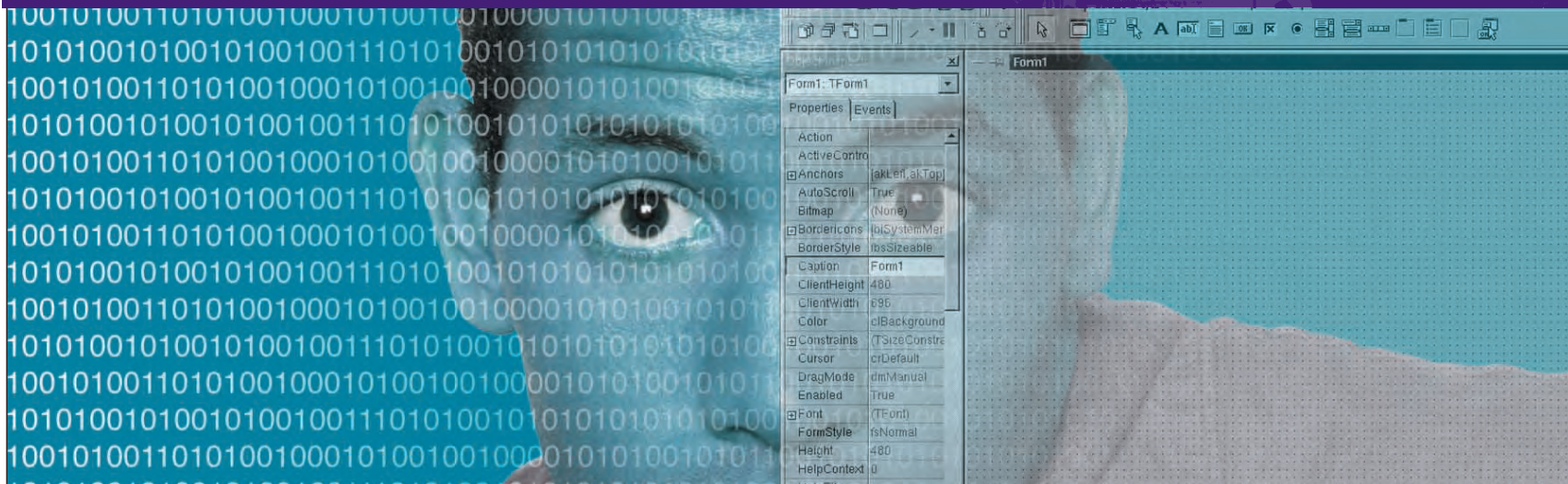
methods:

public String getName()

Returns the name of the method represented by this **Method** object, as a **String**.

public Class getReturnType()

Returns a **Class** object that represents the formal return type of the method represented by this **Method** object.



RAPID APPLICATION DEVELOPMENT

Building a CSV reader

PART 7 This month **Brian Long** shows how the new product info browser for Linux Format cover disks was written

A little while ago our esteemed editor let it be known that Linux Format needed a browser application to supply on each month's cover disk (either CD or DVD format), to allow readers to see what was on the disks and where. The suggestion was that it might be nice to write in *Kyl* if it was up to the job, otherwise it was to be written in C++ with the *GTK* or *Qt* libraries.

Given the challenge we felt obliged to ensure *Kyl* Open Edition got the job done and this month's installment looks at how the finished application (shown in **figure 1**) was built.

Making A Start

The data files that are to be read by the app are stored in CSV (Comma-Separated Values) format. However, every field has quotes around it meaning they can also be read as SDF (System Data Format) files. This is convenient as the *Kyl* **TStringList** class makes it very easy to deal with SDF data.

Each line describes an item on the CD (a tool, library, game,

etc.), for example:

"LXFCD23";"Kyl";"Magazine/Kyl";"Kyl Open Edition again, for you to use with our tutorial series. This directory also contains example files from the tutorial.";

The first field identifies the magazine issue and the cover disk format (the CD version of *Issue 23* in this case), the second is the name of the item. The third field is the location on the CD, the fourth gives a description and the fifth is a URL that provides more information or the most up-to-date version.

The plan is to support passing a file name as a command line parameter. If no appropriate file name is passed the application will look for files with a .csv extension in the current directory. Additionally the user can switch from one CSV file to a different one at any time using a menu item.

Setting Up The Project

The following describes the process to recreate the browser project. In a fresh *Kyl* project, name the form **frmMain**, give it a

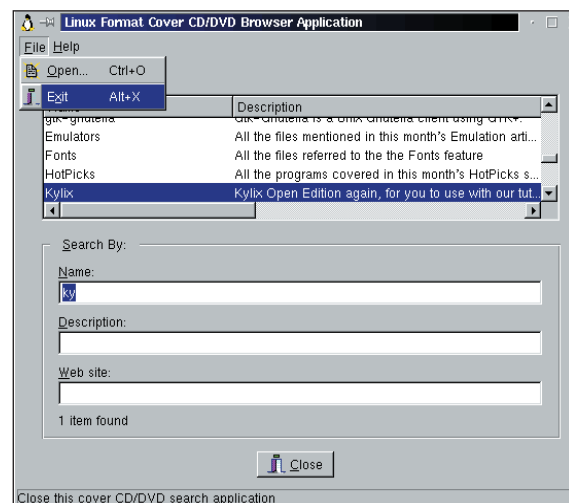
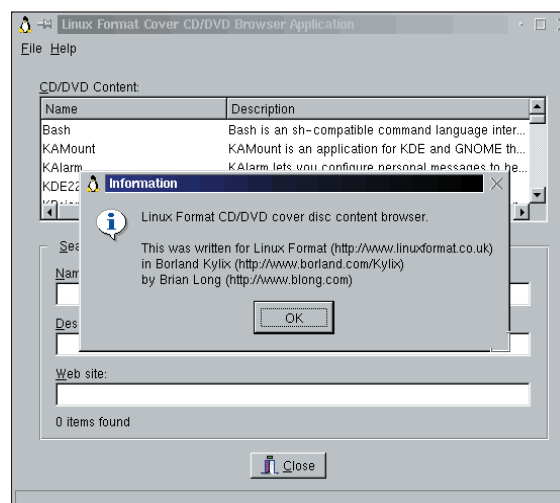


Figure 1: (right) The CSV reader application.

Figure 2: (far right) The CSV reader About box.



descriptive caption and set its Position property to **poScreenCenter**, then add an image list component (from the Common Controls page of the Component Palette) and a main menu component, connecting the two via the menu's Images property.

Set up a menu using the Menu Designer that includes File | Open..., File | Exit and Help | About items called **mniOpen**, **mniExit** and **mniAbout** respectively. Then add some appropriate images to the image list to represent these menu items and associate them with the menu items' **ImageIndex** properties. The event handlers of the latter two menu items are very straightforward (you can see the About box in **figure 2**) and we will come back to the File | Open... menu later:

```
procedure TfrmMain.mniExitClick(Sender: TObject);
begin
  Close;
end;

procedure TfrmMain.mniAboutClick(Sender: TObject);
const
  About = 'Linux Format CD/DVD cover disc content
  browser:#10#10 +
  'This was written for Linux Format
  (http://www.linux-format.co.uk)'#10 +
  'in Borland Kylix (http://www.borland.com/Kylix)'#10 +
  'by Brian Long (http://www.blong.com)';
begin
  MessageDlg('About', About, mtInformation, [mbOk], 0);
end;
```

Next, add a **TBitBtn** (called **btnExit**) from the Component Palette's Additional page and set its **Kind** property to **bkClose** (this gives it an appropriate picture and makes it automatically close the form).

Adding The controls

To allow the CSV file content to be displayed in some easy-to-read way, add a list view component (from the Common Controls page) and name it **lvCDContent**. Set its **MultiSelect** and **RowSelect** properties to **True** (so we can see several selected rows simultaneously) and use the Columns property editor to define the required columns (see **figure 3**).

The column order in the program is: Name, Description, Web Site, Location and Issue (note that this differs from the order in the CSV file). To ensure a nice display set the **AutoSize** property of the Name, Web Site and Location columns to **True** (but not that of the lengthy Description column) and the **MaxWidth** property of Description to **\$8000**.

Next set the **ViewStyle** property to **vsReport** so we can see all the columns and **ShowColumnSortIndicators** to **True** to allow the user to order the list view content by clicking on the various column headers.

To facilitate searching on the more interesting columns add a group box component and place three edit controls inside it, called **edtName**, **edtDescription** and **edtURL**. These will allow the user to enter specific criteria to locate items in the list view.

Adding some text into any, some or all of the edits will reduce the list view to showing only the items that have text matching in the corresponding fields. Since they all do the same thing (specify criteria for different columns) they will ultimately share an **OnChange** event handler to allow one set of code to do the job. Another component to add in the group box is a label (called **lblResults**) with a caption of:

0 items found.

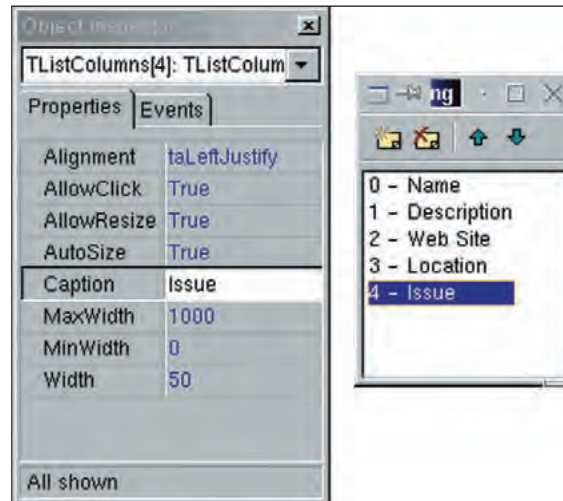


Figure 3: Setting up the list view columns.

This will be updated in the edits' shared **OnChange** handler to reflect how many items met the search criteria.

As an aesthetic touch, and as a help to users of the application, you can drop label components above the edits and list view describing their function. You can also link these labels to their corresponding controls by using an ampersand (&) in their caption and setting their **FocusControl** properties. For example, a label with a caption of **CD/DVD Content:** draws an underscore under the first letter D in its caption. If the user presses Alt plus **D** the label will set focus to whichever control is connected to its **FocusControl** property (the list view for that label).

The final component is a status bar (from the Common Controls page) with its **AutoHint** and **SimplePanel** properties set to **True**, so it automatically shows long hints. Now all that is required to finish the UI is to set up hints for all the controls (you can specify short hints and long hints separated by a pipe sign in their **Hint** property) and set the form's **ShowHint** property to **True** (so all the controls can display their short hints in tooltips). You can see the finished form in the **FormDesigner** in **figure 4**.

To add some polish you can also use the **Anchors** property of the list view, the group box and the edits to make them demonstrate some geometry management and maybe set the form's **Constraints** property to limit how small the form can go (to prevent it being set beyond the limits of usefulness). The pertinent properties in our application are shown in the following list (starts over the page):

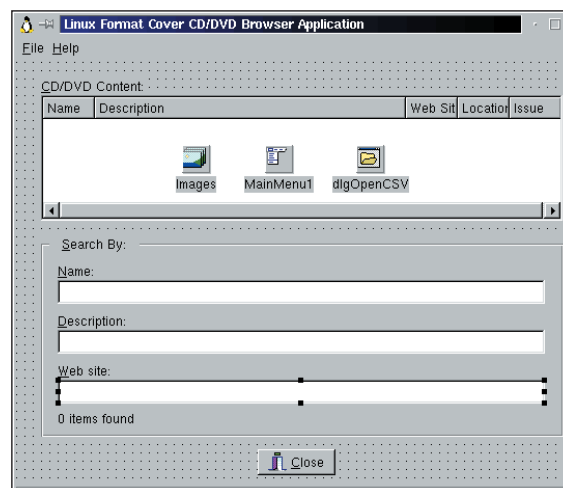


Figure 4: The finished browser UI in Kylix.

LinuxFormatTutorialKylix

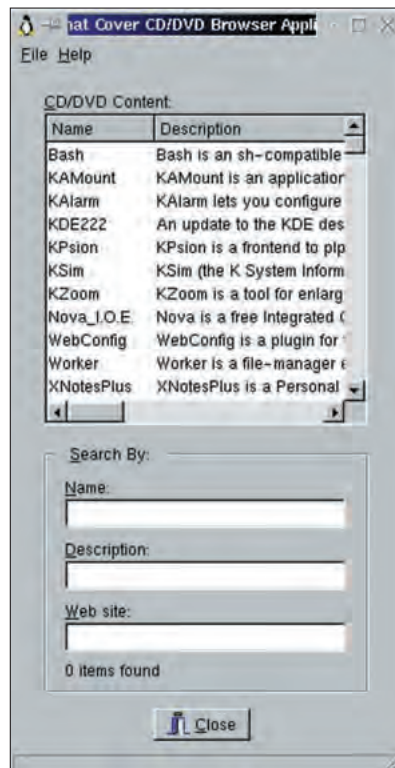


Figure 5: (above) **What the app looks like after resizing, thanks to the handy Anchors property.**

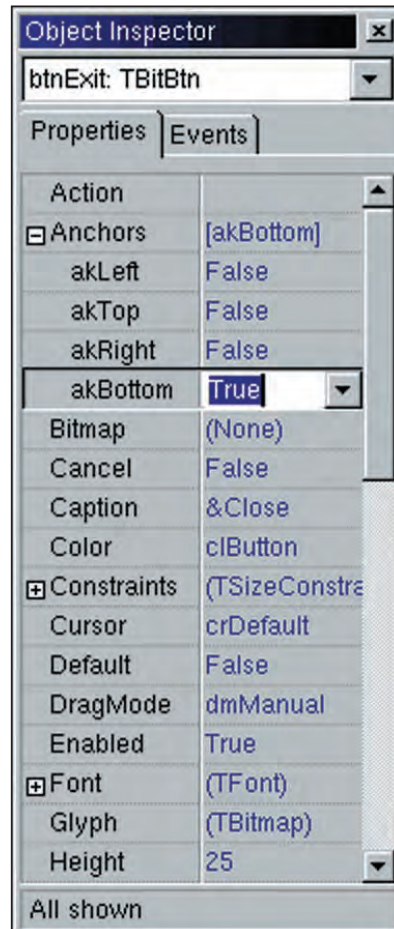


Figure 6: (right) **Setting anchors for the Close button.**

```

frmMain
Width = 552
Height = 451
Constraints.MinHeight = 415
Constraints.MinWidth = 225

lvCDContent
Anchors = [akLeft, akTop, akRight, akBottom]
Hint = CD/DVD contents|This lists everything on this month's
cover CD/DVD'

GroupBox1
Caption = &Search By:
Anchors = [akLeft, akRight, akBottom]

edtName
Hint = Name search|Enter some characters from the Name
field to locate it in the list
Anchors = [akLeft, akTop, akRight]

edtDescription
Hint = Description search|Enter some characters from the
Description field to locate it in the list
Anchors = [akLeft, akTop, akRight]

edtURL

```

```

Hint = Web site search|Enter some characters from the Web
Site field to locate it in the list
Anchors = [akLeft, akTop, akRight]

```

```
btnExit
```

```

Hint = 'Exit application|Close this cover CD/DVD search
application'
Anchors = [akBottom]
Kind = bkClose

```

Anchors

If you haven't played with anchors before you should really do so — they make it a breeze to have controls stretch and contract and/or move as the user resizes the form (see Figure 5).

Normally, a control's Anchors default to [akLeft, akTop] meaning that when the form is resized a fixed distance is maintained between the control's left side and the form's left side and also between the top of the control and the form. Naturally, the gap between the right edges and the bottom edges changes as you resize the form. You can choose other anchor settings to ensure that controls move or resize to keep a fixed or even distance relative to other edges.

In the case of the exit button (which is centred near the bottom of the form), the single anchor value of [akBottom] (see figure 6) makes the button stay just above the bottom of the form and remain equidistant from the left and right edges (as can be seen in figure 5).

Coding The CSV Browser

Having got the application UI out of the way we can now concentrate on how the application is going to work. When the program starts up we need to populate the list view with appropriate items gleaned from the CSV file. In this case we will have two separate methods to do this job; a high level one called **ReadFileIntoListView** and a lower level one called **ReadLineIntoListView**. The former routine takes the name of the file to read as its only parameter.

The form's **OnCreate** event handler will call **ReadFileIntoListView** (and needs to do nothing else). **ReadFileIntoListView** will open the CSV file (whose name is identified by another helper method called **GetCSVFileName**) and for each line call **ReadLineIntoListView**.

The first routine looks like this:

```

procedure TfrmMain.ReadFileIntoListView(const FileName:
String);
var
  List: TStringList;
  I: Integer;
begin
  List := TStringList.Create;
  try
    List.LoadFromFile(FileName);
    //Process each line of file
    for I := 0 to List.Count - 1 do
      ReadLineIntoListView(List.Strings[I])
    finally
      List.Free
    end;
  end;
end;

```

Notice it dynamically creates a **TStringList** object (a convenient object to manage an arbitrary number of strings) to

hold the entire CSV file, which is loaded using the massively helpful **LoadFromFile** method. The actual file name is returned by **GetCSVFileName**, which we will look at in a moment. Each line is in CSV format and holds a record describing a product on the cover disk. It is loaded into the list view by being passed into **ReadLineIntoListView**.

One noteworthy point about a **TStringList** (and also its ancestor class, **TStrings**, from which it inherits) relates to how you read or write an individual string from/to it. The code above shows that you can use the **Strings** array property, however there is an alternative way of doing the same thing. In any class that has string properties, one of them will likely be declared as the *default* array property. If you treat the object itself as if it were an array, the compiler assumes you want to use the default array property; in other words the following two lines of code are functionally identical:

```
ReadLineIntoListView(List.Strings[I]);
ReadLineIntoListView(List[I]);
```

The code for **ReadLineIntoListView** can be found next. This routine takes a string (one line from the CSV file) and writes it to the **CommaText** property of another **TStringList** object (see **figure 7**). **CommaText** is a handy property that does two jobs. When read from, the entire string list is returned in SDF (essentially comma-delimited) format; when written to it accepts an SDF string and splits all the fields into individual lines in the string list. This latter job conveniently turns the CSV line into individual strings, removing all commas and spurious quotes.

```
const
    fldIssue    = 0;
    colIssue    = 4;
    fldName     = 1;
    colName     = 0;
    fldLocation = 2;
    colLocation = 3;
    fldDescription = 3;
    colDescription = 1;
    fldURL      = 4;
    colURL      = 2;

    Columns: array[colName..colIssue] of Byte =
        (fldName, fldDescription, fldURL, fldLocation, fldIssue);

procedure TfrmMain.ReadLineIntoListView(const Line: String);
var
    LineParts: TStringList;
    Item: TListItem;
    I: Integer;
begin
    LineParts := TStringList.Create;
    try
        //Separate string parts into separate lines in TStringList
        LineParts.CommaText := Line;
        //Add text to columns, trimming off excess white space
        Item := lvCDContent.Items.Add;
        //Add in 1st column
        Item.Caption := Trim(LineParts[Columns[colName]]);
        //Add in subitems (remaining columns)
        for I := colDescription to colIssue do
            //Only add if the field exists in the file
            if LineParts.Count > Columns[I] then
                Item.SubItems.Add(Trim(LineParts[Columns[I]]))
            else
                Item.SubItems.Add("");
```

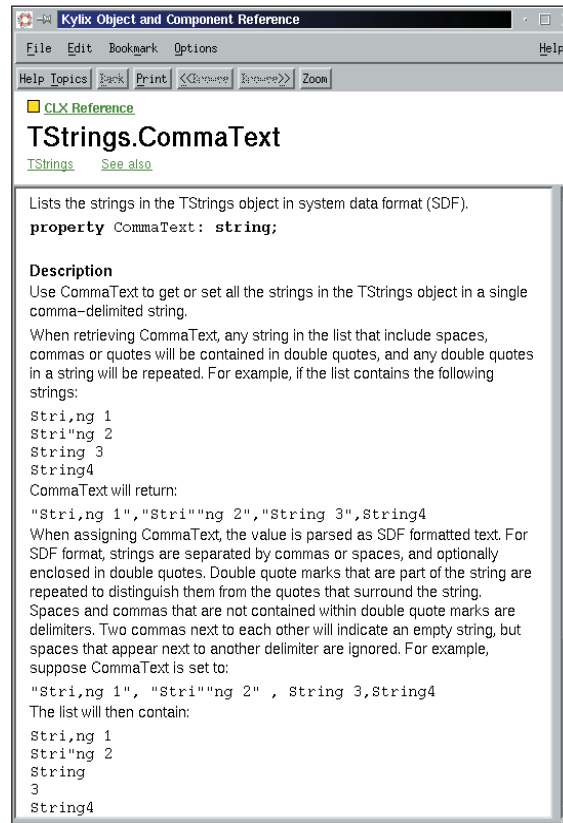


Figure 7: The useful **CommaText** property.

```
finally
    LineParts.Free
end;
end;
```

Having done that a new list item (i.e. row) is added to the list view by calling the **Add** method of its **Items** object property, which we store a temporary reference to in the local **Item** variable. This allows us to access any properties of the list item we might wish to, such as its caption (what's written in the first column) and its **SubItems** property (a **TStrings** object, which we use to add the other columns in).

Again, notice that the columns are not added in the same order as the fields in the file — the order for display starts with the name and description. A number of constants (including a typed array constant) are used to define the file field order and list view column order and translate a column into the corresponding file field.

Choosing A File

Before moving onto the searching logic we should examine how the CSV file name is chosen. We decided that a file name could be passed as a command line parameter to the application, although we need to take care as the application already supports the **-ns** command line switch when compiled with *Kylx Open Edition* (it disables the *Open Edition* nag screen).

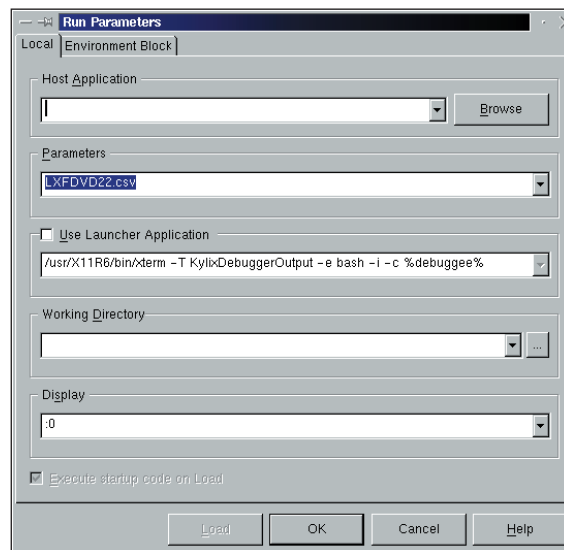
Additionally, if no file name is supplied on the command line the application is designed to look in the current directory for any file with a .csv file extension. The first one found will be used, but if none are found an error message should be generated.

The implementation of **GetCSVFileName** can be found below. You can see the code checks for the **-ns** switch in a case-insensitive fashion (using **CompareText**, which we will see again shortly). Assuming **-ns** is not found, the code checks to



LinuxFormatTutorialKylix

Figure 8: Setting up command-line parameters in the Kylix IDE.



see if any other specified command line parameter exists as a file (the parameter can include a path if required). If a passed file name is located the code jumps straight out of the routine using **Exit**.

```
function GetCSVFileName: String;
var
  I: Integer;
  SR: TSearchRec;
begin
  for I := 1 to ParamCount do
  begin
    Result := ParamStr(I);
    if (CompareText(Result, '-ns') <> 0) and (FileExists(Result))
    then
      Exit; //we found the file (Result), so leave
    end;
    Result := "";
    if FindFirst(*.csv, faAnyFile, SR) = 0 then
    begin
      Result := SR.Name;
      FindClose(SR)
    end
  else
    raise EOpenError.Create('Cannot locate CD/DVD contents
    file');
  end;
end;
```

To test command line parameters in the *Kylix* IDE you should be aware of the Run | Parameters... dialog (see **figure 8**). As you see there is a Parameters field that lets you specify any required parameters (as well as a host of other settings we don't need to worry about right now).

If no file name was passed on the command line the routine moves on to use **FindFirst**. **FindFirst** uses a **TSearchRec** record to keep track of how it got on and is usually used in conjunction with **FindNext**, but in this case we only want the first occurrence. However, assuming **FindFirst** indicates it found a file (by returning **0**) you are obliged to tidy up resources used in the search record by passing it to **FindClose** when you are done.

If no file can be found by searching the current directory the code raises an exception to highlight the problem. We haven't looked at error handling in this tutorial series so far, but we will get there one of these months. This particular syntax is raising an exception that is designed to describe

some sort of file opening problem.

The form's **OnCreate** event handler is shown next. You can see the **GetCSVFileName** routine being called and its resultant file name being passed to **ReadFileIntoListView**. The remaining code specifies which column of the list view should be initially sorted (the disk location column in this case) and sets the **Sorted** property to **True** to act on it.

```
procedure TfrmMain.FormCreate(Sender: TObject);
begin
  ReadFileIntoListView(GetCSVFileName);
  //Match the order in the file by default
  lvCDContent.SortColumn := colLocation;
  lvCDContent.Sorted := True;
end;
```

Searching In The List View

Earlier, whilst building up the UI of the form, we mentioned that the three edit controls would all use the same code to search in the three most interesting columns. Each edit control will use the same event handler for its **OnChange** event so that any key press in any of the edit controls will immediately locate the closest item in the list view.

To set this up double click the first edit control. This makes an **OnChange** handler called **edtNameChange**. To make this event handler look more like a shared event handler, use the Object Inspector's Events page to rename it to **edtSearchFieldChange**.

Now select each of the other edit controls (either individually or both together) and use the drop down arrow on the Events page of the Object Inspector next to the **OnChange** event to share the event handler among the other two events (as shown in **figure 9**).

The code for this shared event handler is quite trivial in that it simply trims any excess white space from around the edit controls' content and passes them to the more involved

LocateItems method:

```
//Shared among the 3 edit controls
procedure TfrmMain.edtSearchFieldChange(Sender: TObject);
begin
  LocateItems(Trim(edtName.Text), Trim(edtDescription.Text),
  Trim(edtURL.Text))
end;

function TfrmMain.MatchFound(const TextEntered, Value:
String): Boolean;
begin
  Result := CompareText(Copy(Value, 1, Length(TextEntered)),
  TextEntered) = 0
  //Result := Pos(UpperCase(TextEntered), UpperCase(Value)) >
  0
end;

procedure TfrmMain.LocateItems(const Name, Desc, URL:
String);
var
  I, Count: Integer;
  Hit: Boolean;
  Item: TListItem;
begin
  //Stop listview doing any screen updates till we are done
  lvCDContent.Items.BeginUpdate;
  Count := 0;
  for I := 0 to lvCDContent.Items.Count - 1 do
  begin
```

```

Item := lvCDContent.Items[I];
//Check for matching name, description or URL, if entered
Hit := False;
if Length(Name) > 0 then
  Hit := MatchFound(Name, Item.Caption);
if Length(Desc) > 0 then
  Hit := Hit and MatchFound(Desc,
Item.SubItems[colDescription-1]);
if Length(URL) > 0 then
  Hit := Hit and MatchFound(URL, Item.SubItems[colURL-1]);
if Hit then
begin
  Inc(Count);
  Item.Selected := True;
  //If this is the first one found, focus it into view
  if Count = 1 then
    Item.Focused := True;
end
else
  Item.Selected := False;
end;
//Make listview refresh its appearance now we are done
lvCDContent.Items.EndUpdate;
if Count = 1 then
  lblResults.Caption := '1 item found'
else
  lblResults.Caption := Format('%d items found', [Count])
end;

```

LocateItems loops through each item in the list view checking each field that the user entered search criteria for, looking for a match. This is done using the **MatchFound** helper function that has two implementations (one is commented out). The active code uses the **CompareText** function to do a simple case-insensitive string comparison to see if the *n* characters entered by the user match the first *n* characters in the field. The **Copy** function is used to extract these characters from the field before comparison. **CompareText** returns **0** if there is a match so **MatchFound** returns the result of a Boolean comparison between the function result and 0 (if a match is found, **CompareText** returns **0** so the Boolean comparison will be True, so **MatchFound** returns **True**).

The inactive version, which you can switch to if you prefer, is more flexible in that it returns **True** if the user's entered text appears anywhere in the CSV field. **Pos** is case sensitive so the two values are upper cased first. **Pos** then returns that character position that the passed in substring appears in the CSV field. *Delphi* strings start at character position 1.

You can find more information about these routines in the help, but note that if you are interested in dealing with international character sets you should use

AnsiCompareText instead of **CompareText** and **AnsiPos** instead of **Pos**.

Notice that the code calls the **BeginUpdate** method of the list view's **Items** object property at the start of the **LocateItems** method and **EndUpdate** at the end. The purpose of this is to tell the list view

that we are planning to do a number of updates to the control and for it not to reflect any of these changes until we are done. Things can get very flickery on the screen if every single change to a control is drawn on the screen — much better to have one redraw after doing what could be a lot of operations with visual side effects.

Each time a match for all appropriate criteria is found the item is selected and a count variable is incremented, otherwise the item is deselected (in case it was selected by a previous search). The first item found is also focused, meaning the list view will automatically scroll it into view, ensuring the user can see at least one selected item.

After each item is checked the results label is updated to show many items were found.

Opening New CSV Files

The final thing to cover is the File | Open... menu item. When chosen it needs to present a dialog to the user allowing them to choose a new CSV file to browse/search through. Assuming the user does not cancel the dialog the list view needs to be cleared and the file read into it in much the same way as the original file.

An open dialog component is required for this (called **dlgOpenCSV**). Set its properties to:

```

DefaultExt = .csv
Filter = Index files (*.csv)*.csv
Title = Open Index File

```

The code for the menu item **OnClick** event handler should look like as follows. The only noteworthy thing in the listing is the last statement, which re-applies the search criteria in the edit controls to the newly loaded file. This is done by directly calling the method that acts as the shared edit event handler. Since this particular event handler pays no attention to the **Sender** parameter (which normally refers to the component whose event is firing) it is okay to pass **nil** as its value.

```

procedure TfrmMain.mniOpenClick(Sender: TObject);
begin

```

```

  if dlgOpenCSV.Execute then
  begin
    lvCDContent.Items.Clear;
    ReadFileIntoListView
(dlgOpenCSV.FileName);
    //Now apply search criteria from edits
    edtSearchFieldChange(nil);
  end
end;

```

Summary

With the application complete and on the cover disk, that's it for this month. The application shows a number of useful *Kyl* techniques: shared event handlers, list views, anchors and constraints (for *Kyl*-style geometry management), command line parameter handling, file finding, text comparisons, tool tips, status bars, string lists and how to handle CSV files (well, okay, SDF files, but they are close enough).


Next month we'll look at more useful *Kyl* techniques. In the meantime, if there is something about *Kyl Open Edition* you want to see covered here, drop us an email and we'll try our best to incorporate it into a future installment. 



Figure 9: Sharing an event handler among different events.

About Brian Long

Brian Long is a UK-based freelance trainer and problem solver for Borland's *Kyl*, *Delphi* and *C++Builder* packages. His Web site is at www.blong.com and he can be emailed at brian@blong.com

KERNEL KNOWLEDGE

The magic SysRq key

A short cut under the hood of your Linux system can be used for good or ill. Hoyt Duff arms you with the facts.

One of the failures in many security schemes can come from the "Security by Obscurity" approach. While the average bloke doesn't know about a security hole, you can be certain the little cretin who wants to trash your system knows them only too well. *Linux Format* will help you close a security hole and learn about a useful tool all in one fell swoop: the Magic SysRq key. We believe that knowledge is power.

You may have never taken much notice of the key labelled 'SysRq' (doubling up with the also useful 'Print Screen', since the days of AT keyboards). Now is the time to sit up and take notice of a direct access route to some of your system's core processes.

The SysRq key functionality is built into the kernel and is intended to be used when the state of the keymap or the kernel is such that you are having serious problems. It's intended as a last-ditch method of controlling your system in some small way if things get out of hand. It can also be used for malicious purposes by any user who has physical access to the machine as they can randomly reboot a production machine or terminate tasks.

The magic of the SysRq key is enabled via **CONFIG_MAGIC_SYSRQ** in the kernel. Most modern pre-compiled kernels already have this function enabled, they just don't tell you about it. Exactly how is it enabled by default? If you have the kernel source code installed, have a look in the file `usr/src/linux/drivers/char/sysrq.c` for the following lines:

```
/* Whether we react on sysrq keys or just ignore them */
int sysrq_enabled = 1;
```

The SysRq key is not without documentation, having a mention in the Keyboard-and-Console HOWTO as well as a text file in the kernel source code documentation. The HOWTO references `usr/src/linux/drivers/char/sysrq.c` as the canonical reference for the SysRq key and identifies `keyboard.h` as the file where the SysRq keystrokes themselves are defined for this role.

For security purposes, it would be wise to modify `sysrq.c` and recompile the kernel to disable this entire feature if you are using Linux to construct a kiosk or system for public use or even for

workstations not under your physical control. If you don't want to recompile, the SysRq functionality can be disabled by root by executing the following, which could also be placed in a startup script:


```
# echo 0 > /proc/sys/kernel/sysrq
```

Also, you can add an entry to the file `/etc/sysctl.conf` as follows:

```
# to disable the sysrq key
```

```
kernel.sysrq = 0
```

Alternatively, specific functions could be disabled or assigned to another key if you wanted to modify the `keyboard.h` file. A clever programmer might also extend the functionality of the SysRq key. And if you are using a non-English keyboard, you'll need to be careful or you may get unexpected results from the key combinations given in the box, below — test first.

Now that you know what it does, see what you can do on your own system to either use it or lose it. 

Web resources

First stop should be the SysRq documentation (if installed) `/usr/src/linux/Documentation/sysrq.txt`
 The Keyboard and Console HOWTO
<http://www.linuxdoc.org/HOWTO/Keyboard-and-Console-HOWTO-8.html#ss8.6>
 The Public Web Browser HOWTO
<http://www.linuxdoc.org/HOWTO/mini/Public-Web-Browser-3.html>

Magic key combinations

How to recover control with SysRq

You may have diddled with the SysRq key itself and concluded that it was disabled and of no use, however the SysRq key is used in combination with other keys in the following manner: You press and hold the Alt (either of them) key and the SysRq key and then press one other key as follows:

B reBoot
E tErminate all tasks using **SIG_TERM**
I kIll all tasks using **SIG_KILL**
K SAK (secure attention Key) and console reset
L kill alL tasks, even *init*
M show Memory information
O power turned Off if hardware allows it
P show Processor registers
R set the Keyboard mode to **K_XLATE** (unraw mode)
U remoUnT all disks read-only
S Sync all disks

T show current Tasks

0-9 any digit sets the kernel logging level and redirects it to the console

You can just imagine how a malicious user can interfere with your system. That the existence of this key is little known is a good thing, but let's examine how to use it for good instead of evil:

If a poorly written app leaves your key in raw mode and you need to return to the unraw mode (where your system interprets keystrokes through a keymap), use Alt-SysRq-R.

If your console is misbehaving, try Alt-SysRq-K to kill everything on that console and reset it.

If your system is hopelessly locked up and you need to reboot it, Alt-SysRq-S-B will sync the disks first and then reboot the system, useful for non-journalled filesystems and removable media.

The P, T and M keys are diagnostic keys and dump info to the console for you to examine.

Coverdisc

Once again **Neil Bothwick** has scoured the world of Linux to bring you an essential selection of the best software, the most useful applications and the tools that will best enhance your Linux life. Enjoy...



Important notice

Before you even put the CD or DVD in your drive, please make sure you read, understand and agree to the following: The *Linux Format* CD/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 CD/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.



Wherever you see this logo it means there's related stuff on the CD

»» READ ME FIRST

This month our CD and DVD have a significant difference. Most of the software we supply on the cover discs is in the form of tarballs, installation packages or ISO images. In order to try it, you have to install it, and some people are reluctant to install every piece of software just to take a look at it. Not so this month — one of our main packages needs no installing to use. The CD and DVD are bootable and take you into a GNUStep desktop environment. This is fully functional, although you can't save configuration settings for obvious reasons.

When you boot from the CD, you'll see a fairly minimal desktop with several large icons. You can start a program by double-clicking the icon, or see the other options by right-clicking it. Read more about Simply GNUstep in the *Distros* box on page 99.

One unfortunate side-effect of making the disc bootable is that the root directory looks rather messy. This isn't a problem if you access the disc via the `index.html` file. If you prefer to use a file manager to browse the disc, the GNUStep directories have lower case names, whereas the normal section outlined on the following pages all have capitalised names.

Desktop

Network hackers may prefer command line based network utilities, but for occasional use a GUI program is easier to get to grips with. We have two versions of the commercial *VisualRoute*. There is a demo of the latest version. This functions as the full version for a 30 day evaluation period. We also have the previous version with no restrictions.

Having a GUI shouldn't mean your keyboard should grow dusty with disuse. Often you don't want to move your hand from the keyboard to mouse just to launch a program. *Xbindkeys* does what the name implies, it allows you to bind commands to keys, so you may launch your favourite programs and utilities without reaching for the mouse. Configuration is by the standard `rc` file, but we also have a *GTK* config editor.

Essential info

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.

Compiling software

A lot of the software on our CD/DVD exists only as source code, usually because that is the only package available, but also because it will then work on any version of Linux, running on any hardware. Thanks to the developer tools `autoconf` and `automake`, the great

majority of source can be compiled in the same way.

1. Copy the source tar file to a suitable location, e.g.

```
cp /mnt/cdrom/developer/fgeneral/fgeneral-0.5.0.tar.gz /usr/src/
```
2. Unpack the source code:

```
cd /usr/src
tar xvfz fgeneral-0.5.0.tar.gz
```
3. Go to the source directory, configure and make the source:

```
cd fgeneral-0.5.0
./configure
make
make install
```

That's it! However, it's always a good idea to check out any `README` files or `INSTALL` files you see in the unpacked source directory, in case there are special options or preferences you might want to change.

Development

If you are writing network software, sooner or later you will need to handle

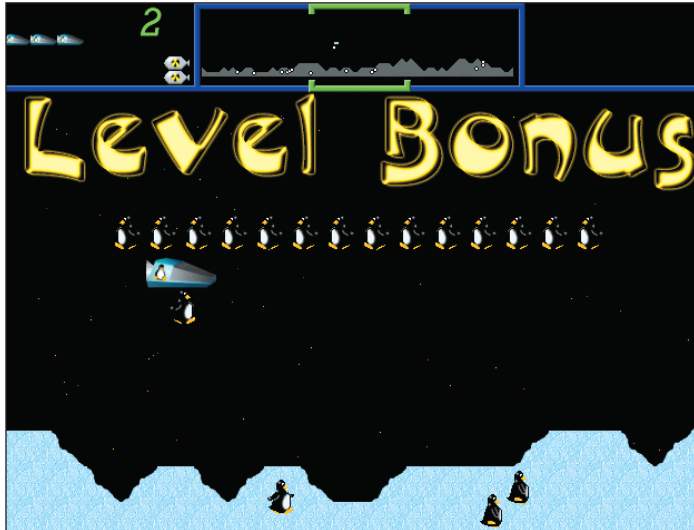
secure connections. *GNU Transport Layer Security Library* implements a secure layer over a reliable transport layer, such as TCP/IP. It currently

supports the Transport Layer Security protocol and SSL (Secure Sockets Layer).

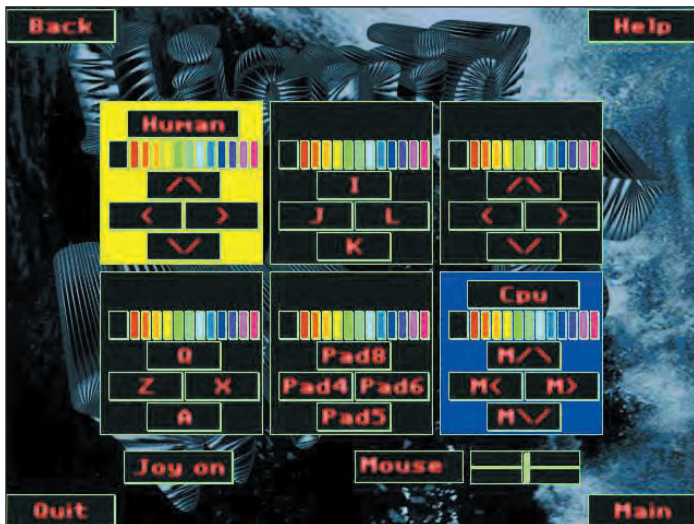
Unless you like reinventing the wheel, a collection like *UDS* should help shorten the time taken to code your programs. *UDS* stands for *Useful Development Stuff*, a collection of objects, functions, classes and dev tools to help you save time.

Games

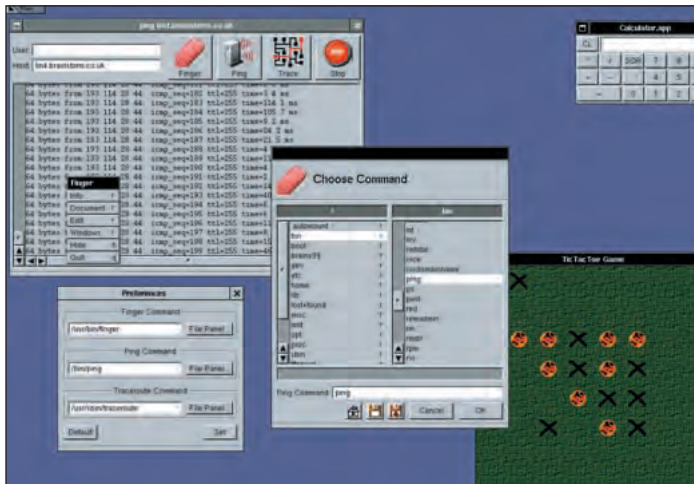
If you long for the good old days of arcades games like *Defender*, it's time for some nostalgia. *Defendguin* is a clone of *Defender*, with one or two changes. I won't spoil the surprise by telling you what they are. If you'd rather play an original game, try *Liquid War*. This is a multiplayer war game. What's different about this? You don't shoot your opponents... you eat them!



Defend the penguins against the evil monopoly.



If you can't beat 'em, eat 'em — *Liquid War* is on.



Try the *GNUMstep* desktop, all you need to do is boot from your coverdisc.

Graphics

If you have any graphics software on your computer, you probably already use it directly, many other programs make use of its features. Because it is so widely used, it helps to keep up to date, so here's the latest version.

Sometimes the urge to play God is simply too great to resist. *Terraform* lets you create your own worlds and landscapes using fractal terrains (also called height fields). This is the sort of program that is rarely used for anything productive, but you can spend *hours* creating landscapes.

Internet

We have two email programs that claim to be fast and lightweight this month. Which is best? Only you can decide that, try out *Balsa* and *CronosII* before making up your own mind. When you have picked your favourite email client, try to choose between the two ICQ programs on the CD, *GnomeICU* and *KXicq2*.

Mention email and thoughts of spam are never far behind. *SpamAssassin* uses several methods to identify unwelcome mails, marking them so you can filter or delete them.

Office

Whether you work in a commercial office, or you just want to keep things

Distros

Most distros take up at least one full CD, often two. Putting them on the CD leaves no room for anything else, so full sized distros will go on the DVD from now on. However, there are several smaller, specialised distros, such as those used for firewalling, that take up far less space.

This month we squeeze on Simply GNUMstep — a distro which aims to be the Mac OS X of the PC. In fact it is based upon the same Objective-C Object Oriented Open Step architecture as OS X and aims to be 100% compatible with the Open Step API. When complete it will be possible to write identical code for Simply GNUMstep and it will run on Cocoa — Apple's update of the Open Step API. Like Redmond Linux, GUI choice has been made for you — the bootable disc will put you straight into a WindowMaker environment, from where you can run various apps directly from the disc, without installation. This is an alpha demo but the work done already is impressive.

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.



LinuxFormatCoverdiscCD

organised at home, we have programs to look after your finances. *Freemoney* is a complete office system, handling customers, products, orders and general ledger, while *KMyMoney2* is aimed squarely at the typical home user.

Server

What happens when an important server goes down, either for maintenance or through a failure? *Fake* can be used to switch in backup servers on a LAN, either manually, or automatically when combined with suitable monitoring software.

There are dozens of logfile analysers for *Apache*, reporting on incoming connections. *Squidalyser* works with logfiles from the *Squid* proxy. This monitors outgoing connections traffic analyser giving overall and per-user reports. These can be used to evaluate bandwidth needs as well as ensuring that your network is not abused.



Track Internet traffic through a GUI — full commercial program.

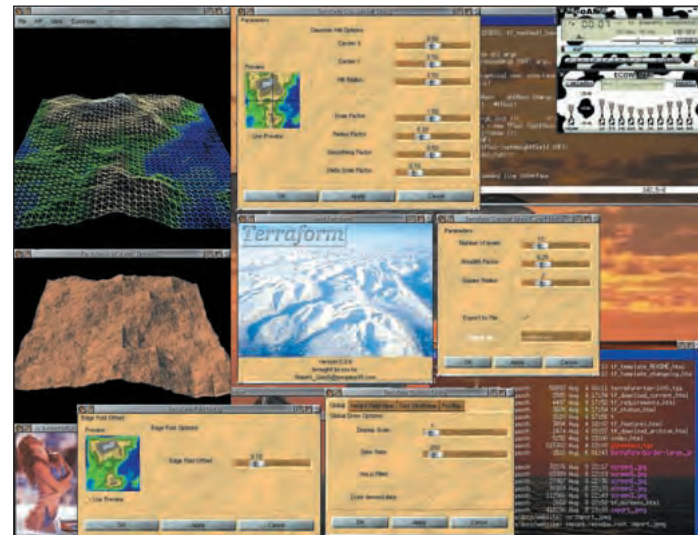
Sound

Audio Transcriber is a program for direct-to-disk recording of large samples. It was written to transfer tapes and LPs to CD, but could equally well be used for radio broadcasts or presentations. *Grip* is a dual purpose program, it plays audio CDs but it also rips tracks from them and optionally encodes to MP3 or Ogg Vorbis.

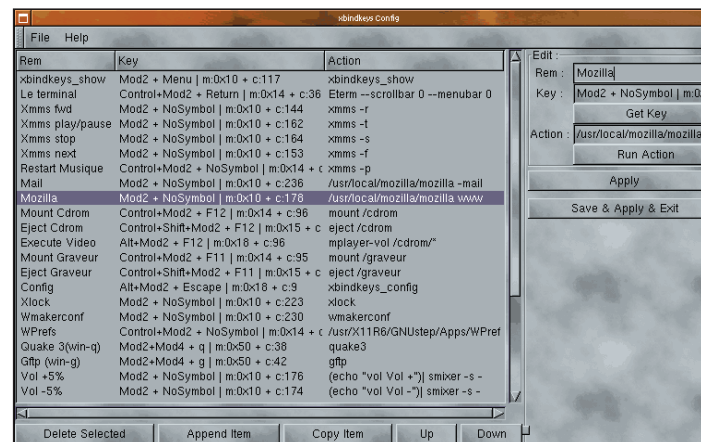
System

XFree86 is a core part of the majority of Linux setups, and it has been updated. Here is *XFree86 4.2.0* as three source tarballs — see the box *Essential info*.

CUPS makes printer handling easier, but things can still get a little tricky if you have more than one printer. The X Printing Panel helps by presenting a list of available printers each time you print — a click on the appropriate one and your document goes exactly where you want it to.



Terraform2 — Design your own scenery, change it if you don't like it, all without lifting a spade.



Assign commands to keys, easily — with Xbindkeys.

CD CONTENTS AT A GLANCE

Magazine

Emulators Files mentioned in this month's Emulation article.
HotPicks Programs reviewed in this month's HotPicks section
PartitionImage Partition Image program, boot disks and libraries
Python Files for the Python tutorial
Roundup CD writing software from this month's round up

Desktop

GRAMPS GTK/GNOME-based genealogy program
KDirStat KDE Directory Statistics
Planets Interactive program for many-body gravity simulations
ScreenSavers Various screensavers
Super-sed sed with a few new features
VisualRoute Visual, fast and integrated ping, whois and traceroute
WuffsMovieDB MySQL+PHP Database catalogue DivXs, VCDs, DVDs etc.
XBbindKeys Associates keys to shell commands under X
XBindKeys_Config GTK program for configuring Xbindkeys
XScreenSaver Modular screen saver and locker

Development

Freddy Programmer's editor using Qt
Gnome-Python Python interfaces to gnome-libs

Katie
Lightning
Luxor
OpenSSLforRuby
TransportLayerSecurity
UDScollecion
XBasic
XForms
RCS, a cross between CVS and NFS
Generates assembly language code at run time
XML User Interface Language toolkit in Java
Binds the OpenSSL library to Ruby
Implements a secure layer over TCP/IP
Useful Development Stuff Collection
Program development environment
Xlib-based GUI toolkit

Games

AfternoonStalker
Defendguin
LiquidWar
MahJong
ScorchedIsland3D
XQF
Clone of the Intellivision game Night Stalker
Clone of the arcade classic "Defender"
Original multiplayer wargame
Play against computer or over network
Variant of Scorched Earth in three dimensions
Quake/QuakeWorld/Quake2/Quake3 server browser

Graphics

ImageMagick
Kandel
PixiePlus
SMPEG
Terraform
VCRonCD
WebMagick
XMovie
Automated and interactive manipulation of images
KDE 2.x program for generating the Mandelbrot set
Browse, manage, and view large numbers of images
General purpose MPEG video/audio player
Create fractal terrains (also called height fields)
Build a bootable CD to play the videos you put on it
Makes image collections available on the Web
Shows high quality Quicktime and MPEG movies

CD and DVD structure

To make it easier for you to find the type of software you are most interested in, the CD and DVD are divided into the following directories:

Magazine

This contains programs and other files mentioned in various articles in each month's magazine. Look here for versions (maybe demos) of reviewed software or programs referred to in tutorials and features.

Desktop

A wide range of programs for general desktop usage of a Linux machine, from full blown windowing environments like *KDE* and *GNOME* to small utilities.

Development

Anything relating to software development. This includes compilers, libraries, classes, debugging tools and development environments

Distros

As you would expect, various Linux distros. These can be full distributions

– only on the DVD – or mini-distros that can also fit on the CD.

Games

What more is there to say, the Games section contains... well, games.

Graphics

Paint, image processing, movie players, video capture. Anything to do with capturing, manipulating or viewing graphics can be found here.

Internet

The Internet directory contains client side internet & network software. This include the obvious candidates like web browsers and email clients as well as some more arcane Internet tools.

Office Various types of productivity software are included here. This is mainly word processors, spreadsheets, databases and accounting software, but there are some other programs that also fit here.

Server

This is the complement of the internet section. It covers everything involved in

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 xjvf /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. 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.

providing services over the Internet or a network, as opposed to using them.

Sound


Once again, a clearly defined category. Anything to do with capturing, playing processing or converting sound or music will go in here.

System

This is where you will find updates and enhancements for various system components as well as various

utilities to improve the usage and security of your system.

Essentials

The Essentials directory contains various programs and packages that are often required by items on Linux Format cover discs but aren't always included as standard with distributions. It also contains the latest kernel sources. The DVD's Essentials directory contains the Linux Documentation Project too. 

Internet

Balsa
CronosII
DNSTrace
Epic4
FtpCube
GnomeICU
KXicq2
LimeWire
rsync
SpamAssassin
WMBiff

Gnome e-mail client with nifty features
Lightweight and fast mail client for **GNOME**
Traces where a Domain Name Server gets its information
EPIC4 is a new direction in ircII development.
Graphical FTP client in Python with GTK bindings
Gnome ICQ client
ICQ client supporting all basic protocol v7 functions
Gnutella client with auto-connect, groups and browsing
Very fast method to bring remote files into sync
Mail filter using several tests to identify spam
Dockapp to display the status of up to 5 mailboxes

Office

Freemoney
KMyMoney2
LinuxTrade
SiagOffice
TkScanFax

Handles customers, products, orders & general ledger
Accounting program aimed at the typical home user
Track prices of stocks and shares
Comprehensive office package
TCL/Tk GUI fax program with support for efax

Server

Cyclone
Fake
PgMarket

The IRC daemon used on SlashNET
Switch in backup servers on a LAN
E-commerce solution built with PHP4 and PostgreSQL/MySQL

MailScanner
Squidalyser
WebMySQL

Email virus scanner and spam tagger
Squid traffic analyser allowing per-user scrutiny
Web interface to a MySQL database server

Sound

AudioTranscriber
CheeseTracker
DivineIntervention
Grip
Kwave
MikMod
Netjuke
Normalize

Records large audio samples to your hard drive
Module music creator similar to Impulse Tracker
Measure and alter the tempo of music
CD player and CD ripper/MP3-encoder for **GNOME**
24-bit sound editor
Curses module player based on libmikmod
Web-based Audio Streaming Jukebox using PHP
Adjusts the volume of audio files to a standard level

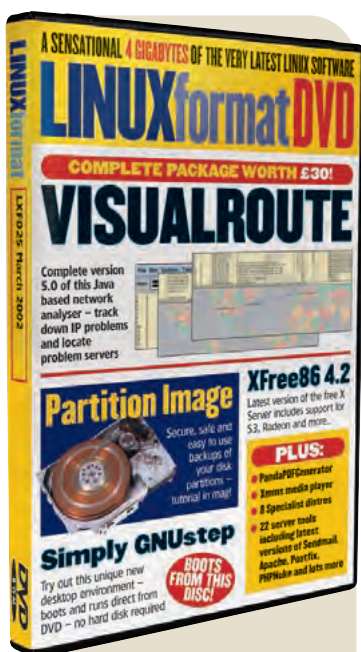
System

BastilleLinux
chkrootkit
GRUB
MakeCDROMRecovery
Net-SNMP
ntop
OpenSSL
Rdiff-backup
Sysmon
XFree86-4.2
XPrintingPanel

Flexible and educational Security Hardening Program
Locally check for signs of a rootkit
Multiboot loader compatible with Linux, Windows & *BSD
Makes a bootable, disaster recovery CD image
Various tools relating to **SNMP**
Shows network usage like top does for processes
Fully featured, and Open Source SSL toolkit
Backs up one directory to another
Network monitoring tool covering many services
The latest version of XFree86 — 4.2.0
Choose the desired printer at print time

DVD

Neil Bothwick is your guide through the wonders of this month's jam-packed **Linux Format DVD**. The distros and dev tools should keep you busy, but don't forget the games and desktop toys.



Important notice

Before you even put the CD in your drive, please make sure you read, understand and agree to the following: The *Linux Format* CD 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 CD 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.



Wherever you see this logo it means there's related stuff on the DVD

Distros

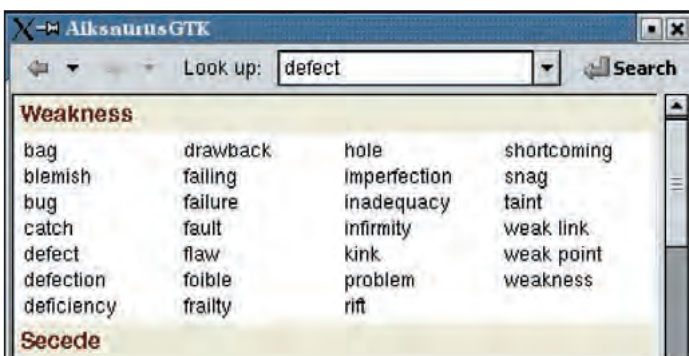
In recent months we have brought you the latest distributions from Red Hat and Mandrake. On this DVD we have all the updates to both distributions since their release. You should download and install security updates as soon as they are released, but other updates are less urgent and this can save you some serious download time. The Mandrake directory contains

the regular updates, not the bleeding edge ones from the Mandrake Cooker.

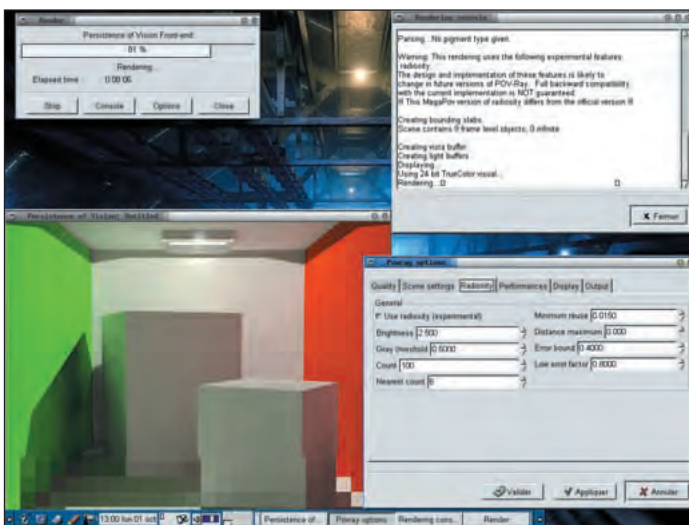
Instead of one or two major distributions, we have several small ones this month. These are more specialised distributions, for tasks such as routing or system rescue.

Desktop

CD users get the chance to create their own landscapes on Earth, but



Stuck for words? *Aiksaurus* may be able to help.



TrueVision is a modeller for *POV-Ray*, which is also on the DVD.

Defective CDs. In the unlikely event of your CD/DVD being physically damaged we'll send you a new, working version within 28 days. Send your defective disc – complete with your name, address, and a description of the fault – to: **LXF Disk Returns, TIB Plc, Unit 5, Triangle Business Park, Pentrebach, Merthyr Tydfil, Mid Glamorgan CF48 4YB.**

DVD users have the opportunity to colonise Mars. The *Mars Simulation Project* creates a simulation of a future human settlement on the planet Mars.

Every month the Linux format DVD contains ISO images of bootable distribution CDs, so you need something to burn them. *CDBakeOven* provides a user-friendly, full-featured CDR(W) burning facility.

Development

If the recent inclusion of Linux From Scratch piqued your interest in DIY distros, *gBootRoot* may well be of interest. It is designed to make development and testing of distributions "fun and simple".

Lrczo is a library of network functions. It provides network functionalities for Ethernet, IP, UDP, TCP, ICMP, ARP and RARP protocols, making the development of network software that much easier.

Internet

In addition to the two email clients on the CD, the DVD also has *Mahogany*. This is a cross-platform mailer that has its own Python interpreter to add scripting facilities. Don't put others through "404 error" misery, use *Link checker* to make sure every link on your web site points to an existing page.

Server

With three mail servers, two FTP servers, two web forum suites, a news server and a mailing list manager, this directory is essential browsing for anyone running any kind of server. Oh, and the latest *Apache* is in here too.

There's more

Open Office has been on a Linux Format DVD before, but there have been several major updates since then, so it is well worth another look.

» DVD CONTENTS AT A GLANCE

Desktop

Aiksaurus	English language thesaurus library and GUI
CDBakeOven	User-friendly, full-featured CDR(W) burning facility
CDfs	File system that 'exports' all tracks on a CD
Dav	Stable and efficient text editor
DirectVNC	VNC client using the DirectFB library
FreeTTS	Speech synthesis system in Java
KAlarm	Display personal messages at scheduled times
LyX	Open source document processor
MarsSimulationProject	Simulate a future human settlement on the planet Mars
PowerJournal	Easy-to-use journal/diary application
PWgui	Graphical tool to test the strength of passwords
SANE	Scanner Access Now Easy
SANEFrontends	Frontends to SANE ("Scanner Access Now Easy")
SimpleCDR-X	GTK+ GUI for CD writing, mastering and manipulation
SpaceChart	GNOME application that shows the stars in space in 3D
Stellarium	Renders 3D photo-realistic skies in real time
Waimea	Virtual desktop window manager
WaimeaMiniPager	Dockapp pager for the Waimea window manager
WindowMaker	Window manager for the GNUstep Desktop Environment
wmnetload	Network interface monitor dockapp for Window Maker
wmSolar	Shows you the solar system viewed from top
X10ephem	Computes sunrise and sunset times
Xfce	Easy to use and configure X11 environment

Development

AESforRuby	Compact and fast AES encryption for Ruby programs
FLAC	Free Lossless Audio Codec
FLTK	GUI toolkit for X11, Windows, MacOS X and OpenGL
FormProc	Java library for handling and validating forms
gBootRoot	Construct and develop distributions
GNet	A simple network library
Graphviz	A set of graph drawing tools and libraries
lcrzo	Library for network admins and network hackers
OpenMotif	Latest version of this standard GUI
PandaPDFGenerator	Makes PDFs on the fly for the Web or general use
ProjectCenter	Project manager for GNUstep
PyGTK	Set of Python bindings for the GTK widget set
PythonXML	Parsers, SAX and DOM interfaces, with sample programs
SDLsound	Library to decode several sound file formats
Syslinux	Boot loader for Linux
wxBasic	Cross-platform Basic interpreter

Distros

BasicLinux	Mini-Linux that boots from a DOS/Win9x partition
ClarkConnect	Turns an old PC into a smart, secure Internet gateway
CRUX	i686-optimised distribution for experienced users
FloppyFW	Router and simple firewall on a single floppy
FloppyGate	Distro on a floppy to share an Internet connection
Herbix	FTP, HTTP, IRC, DHCP and SMTP server on a floppy
IPCop	Distro to protect the networks it is installed on
MandrakeUpdates	The latest updates for Mandrake 8.1
RedHatUpdates	The latest updates for Red Hat 7.2
ROOTLinux	Advanced Linux distro, not recommended for newbies
Wolverine	Firewall and VPN server based on Embedded Coyote

Games

AirTraffic	Puts you into an air traffic controller's hotseat
DooMLegacy	Enhanced port of Doom using OpenGL
DungeonCrawl	Fun game in the tradition of Rogue, Hack, and Moria
Euchre	Spades or bridge-like card game played with four players
FindItNIX	Clone of the Windows game FindItXP
StarWar	2D space shooting game similar to xsoldier
WarsOfDestiny	Real time strategy game
XLightOff	Easy to play, harder to solve.
Xword	Open and solve AcrossLite puzzle format files

Graphics

DFBSee	Image viewer and video player that uses DirectFB
---------------	--

Gmerlin
Grabbo
MPlayer
POVRAY
Showimg
Truevision

Internet

Evolution
Galeon
gFTP
KLCC
LinkChecker
Mahogany
Opera
Pan
POP3spy
Wget

Office

DataVision
eXist
Gnumeric
NOLA
OpenOffice
Scribus

Server

Apache
ApacheToolbox
Arbomb
Courier
DHCPcd
DNews
FUDforum
ISC_DHCP
MySQLNavigator
PHPNuke
phpPgAdmin
Postfix
ProFTPD
PureFTP
Sendmail
Sympa

Sound

Audacity
Cajun
ColdFeetCajun
COW
Ecasound
LAME
LibVorbis
SDLmixer
TerminatorX
Trommler
VorbisTools
XMMS

System

DebFoster
DumpRestore
Etherboot
Ethereal
Fink
FSBackup
Linuxconf
LinuxTestProject
OpenPKG
Parted
TimosRescueCDSet
Webmin


Audio/video player with plugins and a playlist
Real-time video grabber based on GtkVCL
Movie and animation player for many file formats
The Persistence Of Vision Raytracing Toolkit
Feature-rich image viewer handling numerous formats
3D modeler for povray with integrated GNOME frontend

Mailer, contact manager and communications tool
Efficient Web browser using the Mozilla engine
Multithreaded FTP client using GTK+
KDE client for the LineControl server
Check HTML documents for broken links
Cross-platform GUI email client
Alternative, lightweight and fast Web browser
Newsreader, loosely based on Agent and Gravity
List, read and delete mails for a POP3 mailbox
Non-interactive download of files from the Web

Database reporting tool similar to Crystal Reports
XML database with pluggable storage backends
Powerful and easy to use spreadsheet program
PHP program for accounting and inventory management
Open Source version of the StarOffice productivity suite
Easy-to-use tool for simple Desktop Publishing

The world's most popular HTTP server
Easily compile many Apache modules
Detects volatile archive files used in DoS attacks
An integrated mail/groupware server
DHCP client daemon
Install your own local news server
PHP & MySQL forum with many customizable features
DHCP client, server and relay agent
MySQL database server GUI client program
Web portal and online community system
Port of phpMyAdmin for PostgreSQL
An alternative to the widely-used Sendmail
Proven, high-performance, scalable FTP server
Fast, standards-conformant FTP server
The de facto standard Mail Transfer Agent
Scalable and highly customizable mailing list manager

Cross-platform multitrack audio editor
Turn any computer into a massive audio jukebox
Rewritten version of Cajun
Generates a wave file from one or more graphs
Software package for multitrack audio processing
Command line MP3 Encoder
High-quality lossy audio codec
Simple multi-channel audio mixer
Real time audio synthesizer
X-based drum machine with real time output
Tools for the Ogg Vorbis audio codec
Multimedia player based on the look of WinAmp

Detects "orphaned" packages and cleans them up
The standard backup and restore tools
Makes ROMS for booting x86 PCs over a network
Network protocol analyser, or "packet sniffer"
Downloads, build and installs source tarballs
Incremental backup creation utility
Sophisticated system administration tool
A collection of tools for testing Linux
Flexible and powerful software packaging facility
Create, destroy, resize, and copy partitions
Makes bootable CDs with a ready to use rescue system
Web based interface for system administration 

User Groups

Your local Linux User Group needs you! LUGs worldwide are full of members keen to help with your problems, discuss ideas and generally natter about all things Linux. We have collected a load of information here so you can find the LUG closest to you. You can find lots more information online at: www.lug.org.uk or at www.linuxformat.co.uk/links.php

1 Hampshire

URL www.hants.lug.org.uk
Contact Hugo Mills

2 Bristol & Bath

URL www.bristol.lug.org.uk

3 Scottish

URL www.scottish.lug.org.uk
Contact Tony Dyer

4 Oxford

URL www.oxford.lug.org.uk
Contact Alasdair G Kergon

5 Bromcom (Kent)

URL www.kent.lug.org.uk
Contact John Mills

6 Brighton

URL www.brighton.lug.org.uk
Contact Johnathan Swan

7 Sussex

URL www.sussex.lug.org.uk
Contact Mike Pedley

8 Northants

URL www.northants.lug.org.uk
Contact Kevin Taylor

9 Anglian

URL www.anglian.lug.org.uk
Contact Martyn Drake

10 Milton Keynes

URL www.mk.lug.org.uk
Contact Denny De La Haye

11 Doncaster

URL www.doncaster.lug.org.uk
Contact Andy Smith

12 Moray

URL www.moray.lug.org.uk
Contact Stewart Watson

13 West Wales

URL www.westwales.lug.org.uk
Contact Dan Field

14 Wolves

URL www.wolves.lug.org.uk
Contact Jono Bacon

15 Peterborough

URL www.peterboro.lug.org.uk
Contact Steve Gallagher

16 Edinburgh

URL www.edinburgh.lug.org.uk
Contact Alistair Murray

17 Tyneside

URL www.tyneside.lug.org.uk
Contact Brian Ronald

18 Leicester

URL www.leicester.lug.org.uk
Contact Clive Jones

19 Greater London

URL <http://gl.lug.linux.co.uk/>

20 Surrey

URL www.surrey.lug.org.uk
Contact Jay Bennie

21 Cambridge

URL www.cam-lug.org

22 Devon & Cornwall

URL www.dclug.org.uk/
Contact Simon Waters

23 Falkirk

URL www.falkirk.lug.org.uk

24 Manchester

URL www.manlug.mcc.ac.uk
Contact John Heaton, Owen Le Blanc

25 Hertfordshire

URL www.herts.lug.org.uk
Contact Nicolas Pike

26 West Yorkshire

URL www.wylug.lug.org.uk
Contact Jim Jackson

27 Sheffield

URL www.sheflug.co.uk
Contact Richard Ibbotson

28 Staffordshire

URL <http://www.staffslug.org.uk/>

29 North East

URL www.shofaruklinux.net/NELUG

30 London

URL www.lonix.org.uk

31 Thames Valley

URL www.sclug.org.uk

32 Liverpool OpenSource

URL http://linux.liv.ac.uk/_liv_linux_ug/
Contact Simon Hood

33 Deal Amiga Club

Email superhighwayman@hotmail.com
Contact John Worthington

34 Chesterfield

Email spirelug@yahoo.co.uk
Contact Robin Needham

35 South Derbyshire

URL www.sderby.lug.org.uk/
Contact Dominic Knight

36 Belfast (BLUG)

URL www.belfastlinux.cx
Contact Ken Guest

37 Wiltshire

URL www.wiltshire.lug.org.uk
Contact Jason Rudgard

38 South London

URL www.sl.lug.org
Contact Ben@benguin.co.uk

39 Cheshire

URL www.sc.lug.org.uk
Contact Anthony Prime — enquiry@sc.lug.org.uk

40 North Wales

URL www.northwales.lug.org.uk
Contact Jonathan Cole

41 Midlands

URL www.midlandsLUG.cjb.net **WARNING: Popup ads**
Contact Pete Thompson

42 Cumbria

URL www.cumbria.lug.org.uk
Contact Jamie Dainton

43 Dorset

URL www.dorset.lug.org.uk
Contact Beanz and Tracy

44 Shropshire

URL www.shropshire.lug.org.uk **WARNING Flash used on this site, may also trap Netscape's back button!!!**
Email shropshire@lug.org.uk

45 South West

URL www.southwestlug.uklinux.net/
Email southwest@lug.org.uk

46 South Wales

URL www.sw.lug.org.uk
Contact Tim Bonnell

47 North London

URL <http://www.kemputing.net/alt/lug/anlug.html>

48 Malvern

URL www.malvern.lug.org.uk
Contact Greg Wright

49 Huddersfield

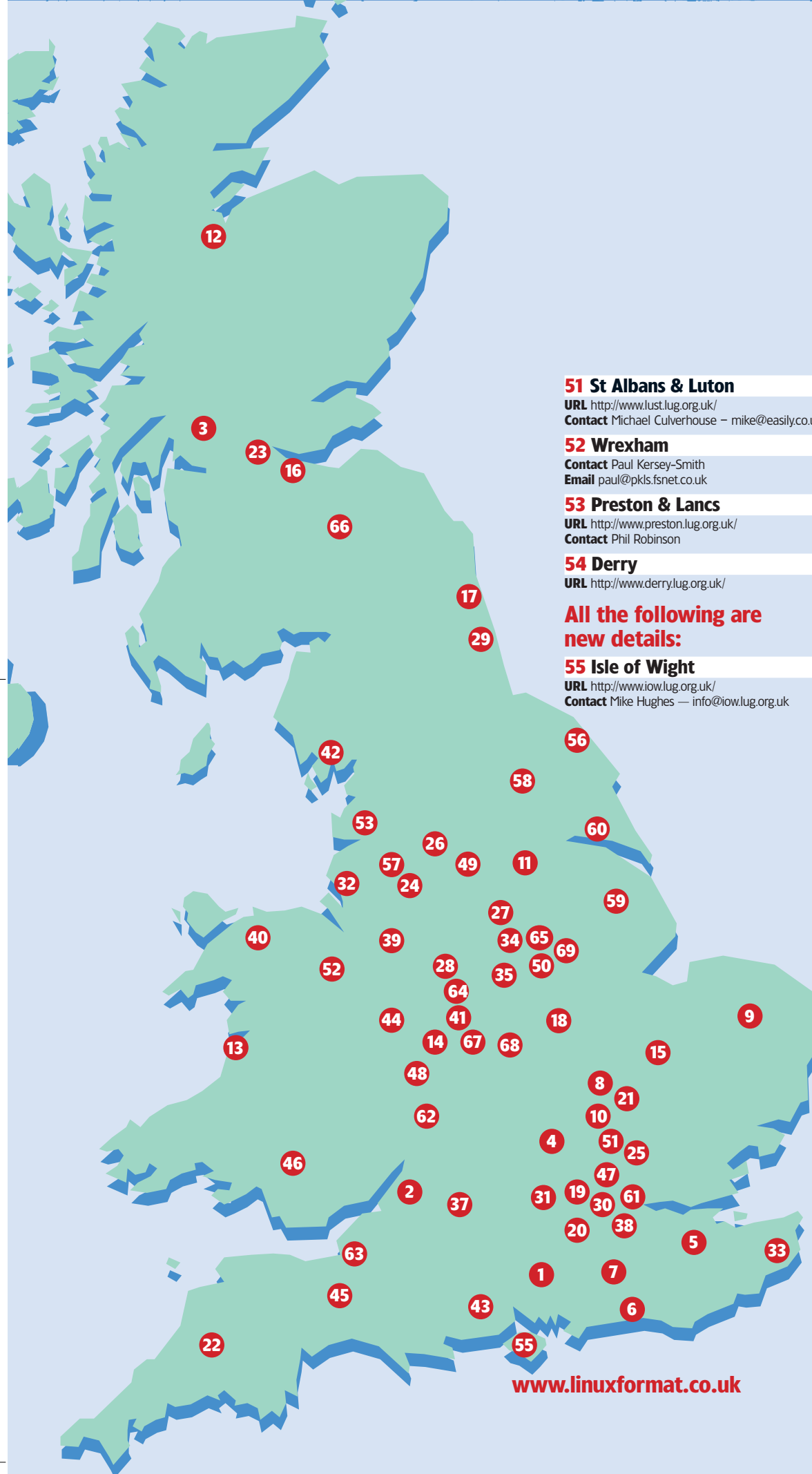
URL www.hud.lug.org.uk
Contact Adam Brookes

50 Nottingham

URL www.nottingham.lug.org.uk
Contact Godfrey Nix

REVISED
DETAILS

REVISED
DETAILS

**51 St Albans & Luton**URL <http://www.lust.lug.org.uk/>Contact Michael Culverhouse — mike@easily.co.uk**52 Wrexham**

Contact Paul Kersey-Smith

Email paul@pkls.fsnet.co.uk**53 Preston & Lancs**URL <http://www.preston.lug.org.uk/>

Contact Phil Robinson

54 DerryURL <http://www.derry.lug.org.uk/>

**All the following are
new details:**

55 Isle of WightURL <http://www.iow.lug.org.uk/>Contact Mike Hughes — info@iow.lug.org.uk**56 Scarborough**URL <http://www.scarborough.lug.org.uk/>

Contact Jamie Adams

57 Blackburn

Contact Matt Wright

Email matt@consultmatt.co.uk**58 York**URL <http://www.york.lug.org.uk/>

Contact Ewan Mac Mahon

59 LincsURL <http://www.lincs.lug.org.uk/>

Contact Chris Lingard

60 HullURL <http://www.hull.lug.org.uk/>Contact Chrus Burton chris@hull.lug.org.uk**61 East London**URL <http://www.eastlondon.lug.org.uk/>

Contact Jonathan Spriggs

62 Gloucestershire & CotswoldsURL <http://www.gloucs.lug.org.uk/>

Contact Barrie Haycock

63 Yeovil CollegeURL <http://www.yclug.lug.org.uk/>

Contact Adam Parker

64 South StaffordshireURL <http://www.staffs.lug.org.uk/>

Contact Oliver Keenan

65 MansfieldURL <http://www.mansfield.lug.org.uk/>

Contact Brent Vardy

66 BordersURL <http://www.linux.bordnet.co.uk/>

Contact Welby McRoberts

67 South BirminghamURL <http://www.sb.lug.org.uk/>

Contact Tim Williams

68 Coventry

Contact Darren Austin

Email info@coventry.lug.org.uk**69 Newark**URL <http://www.newlinc.lug.org.uk/>

LinuxUserGroups

LUG OF THE MONTH!

York

The York Linux Users Group is a relatively young but fast growing LUG, founded a little over a year ago. Meetings are held at or near the University of York, and while staff and students attend, the LUG is open to all and the membership includes plenty of non-university people.

Meetings occur at least once every four weeks, and

typically include a talk given by a brave volunteer YLUG member followed by drinks and a chat in a nearby bar. During university terms minor meetings are also held between the major meetings, and feature just the drink and chat.

Much of YLUG's day to day activity takes place on our lively and friendly mailing

list, details of which are on our website, which is currently being revamped and expanded:

<http://york.lug.org.uk/>

You can email us your submission at linuxformat@futurenet.co.uk (please use the subject "LUG details" or something similar), or alternatively by post to: LUG Info, Linux Format, 30 Monmouth Street, Bath BA1 2BW.

Hope to hear from you!



Worldwide Linux User Groups

Free software users across the globe.

Africa

PRETORIA

URL www.plug.za.org

Email andriesn@icon.co.za

STELLENBOSCH

URL www.entropysun.ac.za/

Email ixion@entropysun.ac.za

Australia

ADELAIDE LUG

URL www.linuxsa.org.au

Email mtippet@anu.edu.au

MELBOURNE, VICTORIA

URL www.luv.asn.au

Contact luv-committee@luv.asn.au

PERTH

URL plug.linux.org.au

Europe

AUVERGNE

URL www.linux-arverne.org/

Email Cyril.Hansen@wanadoo.fr

EIRE

URL www.linux.ie

Email root@linux.ie

URL www.dilu.org

Email glossary@dilu.org

GHENT

URL lsgg.rug.ac.be/

Email wvdputte@lsgg.rug.ac.be

GOTHENBURG

URL nain.oso.chalmers.se/LUGG/index.html

LISBON

URL www.students.iscte.pt/~a12593/gul.html

Email Paulo.Trezentos@iscte.pt

India

URL www.river-valley.com/tux/index.html/

Email anil@river-valley.com

URL www.linux-india.org

Email newsmaster@linux-india.org

UK

Don't forget the distribution-specific mailing lists:

URL <http://www.lug.org.uk/maillist.html>

North America

ALASKA

URL www.aklug.org/index.html

Email deem@wdm.com

BATON ROUGE

URL www.brbug.net/

Email dpuryear@usa.net

BAY AREA

URL www.balug.org/

Email aftyde@balug.org

CLARKSVILLE, TN

URL <http://www.clug.org>

Email tux@clug.org

DENVER

URL spotelfwerks.com/~clue/

Email: lynnd@ihs.com

LOS ANGELES

URL www.lalugs.org/

Email dank@alummi.caltech.edu

NORTH COLORADO

Email nclug@nclug.org

Contact Mat Taggart

TAMPA

URL terrym.com/slug/index.html

Email paulf@quillandmouse.com

UHACC Normal, IL

URL <http://www.uhacc.org/>

Email lug@uhacc.org

VIRGINIA TECH

URL corvette.me.vt.edu/pages/index.html

Email nega@vt.edu

South America

BUENOS AIRES

Email dcoletti@impost.com.ar

LIMA

URL linux.unired.net.pe/

Email linux@unired.net.pe

MONTEVIDEO

URL www.linux.org.uy/

PARAGUAY/ ASUNCION

Email rolgiati@conexion.com.py

SAO PAULO

URL gul.linux.ime.usp.br/

Email gul@ime.usp.br

Linux User Groups

Welcome to our expanded User Group pages. With nearly 70 LUGs active around the UK, and several hundred more around the world, we just had to give them more space.

LUGs are often the focal point for Free Software involvement in a community, organising events for local businesses, helping schools and even providing a good excuse to go down the pub and meet some friends.

LUGs need you!

However the one thing a LUG needs to thrive is *you*. A LUG is a group of Linux Users – no more, no less. The success of each and every LUG is down to the enthusiasm of its members. It needn't involve too much effort – just answering the odd question on the LUG mailing list, or helping to keep the website up to date – but each and every member counts.

If you want to support Free Software use in your local area then joining your local LUG is a great start. If you live in the UK have a look at <http://www.lug.org.uk/lugs/index.html> If not take a look at <http://www.linux.org/groups/> We will try and keep our map (see *page 109*) as up to date as possible, but there is always the chance that a new LUG has appeared after we have gone to press.

As well as helping your fellow Linux users, the chances are that you will benefit from the different experiences of your fellow members.

LUG events

Last year's Linux Day showed the potential both for LUG events and company involvement. LUGs can promote Linux in a variety of ways. Sheffield LUG have kicked off this year with a huge Linux seminar involving IBM and SuSE, along with local businesses and a speaker from the Free Software Foundation Europe. If you run a business, then why not consider sponsoring an event?

What, no LUG?

If there is no LUG in your area, then there is only one thing for it – start your own. As with all things Linux there is a HOWTO guide at <http://www.linuxdoc.org/HOWTO/User-Group-HOWTO.html> As well as a short guide at <http://www.lug.org.uk/lugmasters/howto.html>

Linux User Group organisers

If you're not listed here, or we have your details wrong, please contact us at: **LUGS!, Linux Format, 30 Monmouth Street, Bath, BA1 2BW** or email your details to: linuxformat@futurenet.co.uk

EDITORIAL**Editor** Nick Veitch nick.veitch@futurenet.co.uk**Art Editor** Julian Jefferson julian.jefferson@futurenet.co.uk**Reviews Editor** Richard Drummond richard.drummond@futurenet.co.uk**Production Editor** Richard Smedley richard.smedley@futurenet.co.uk**Editorial Contributors**

Jono Bacon, Neil Bothwick, Andy Channele, David Cartwright, David Coulson, Hoyt Duff, Simon Goodwin, Bill von Hagen, Maurice Kelly, Jon Kent, Martin Linklater, Brian Long, Biagio Lucini, John Mehaffey, Charlie Stross

COVER CD PRODUCTION**CD Editor** Neil Bothwick**Senior CD Editor** Kate Hadley**ART CONTRIBUTORS****Photography** Rick Buettner, Corbis, Jeff Moore, Superstock, Amanda Thomas**Illustration** Paul Bateman, Shane Collinge, Chris Winn**ADVERTISING SALES**

Matt Stanley 01225 442244

matt.stanley@futurenet.co.uk

Emma Mimmack 01225 442244

emma.mimmack@futurenet.co.uk**MARKETING AND PROMOTIONS****Marketing Manager** Fiona Tully**Subscriptions Manager** Nick Skardon**PRODUCTION****Production Co-ordinator** Diane Ross**Group Production Manager** Clare Tovey**Advertising Production Co-ordinator** Jo Crosby**MANAGEMENT****Publisher** Dave Taylor**Group Advertising Manager** Simon Moss**Group Publisher** John Weir**Circulation Director** Sue Hartley**Managing Director** Colin Morrison**DISTRIBUTION AND CIRCULATION****Circulation Manager** Jamie Malleyjamie.malley@futurenet.co.uk**Distributed by** Seymour Distribution, 86 Newman Street, London W1T 3EX**Tel** +44 (0)207 3968000**Overseas Distribution by** Future Publishing Ltd.**Tel** 01225 442244.**Overseas Licences**Simon Wear simon.wear@futurenet.co.uk**Tel** +44(0)1225 442244 **Fax** +44 (0)1225 732361**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 96**Email** linuxformat.subs@futurenet.co.uk

Copyright No part of this publication may be reproduced without written permission from our publisher. We assume all letters sent – by email, fax or post – are for publication unless otherwise stated, and reserve the right to edit contributions. All contributions to Linux Format are submitted and accepted on the basis of non-exclusive worldwide licence to publish or license others to do so unless otherwise agreed in advance in writing recognises all copyrights and trademarks. Where possible, we have acknowledged the copyright holder. Contact us if we haven't credited your copyright and we will always correct any oversight.

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. We cannot be held responsible for mistakes or misprints. Linux Format recognises all copyrights in this issue. Where possible we have acknowledged the copyright holder. Please contact us if we have failed to credit copyright.

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 © Future Publishing Ltd 2002

LINUX is a trademark of Linus Torvalds, GNU/Linux is abbreviated to Linux throughout for brevity.

All other trademarks are the property of their respective owners

Future Publishing Ltd. is part of The Future Network plc.

The Future Network produces carefully targeted specialist magazines and websites for groups of people who share a passion. We aim to satisfy their passion by creating titles that offer superb value for money, trustworthy information, multiple ways to save time and money, and are a pleasure to read or visit. Today we publish more than 80 magazines and over 30 magazine websites and networks from offices in five countries. The company also licenses 32 of its titles resulting in over 60 local editions in a further 23 countries. 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**Chief Operating Officer & Managing Director, UK:** Colin Morrison**Group Finance Director:** John Bowman**Tel** +44 (0)1225 442244**www.thefuturenetwork.plc.uk****Media with passion**

BATH • LONDON • MILAN •
NEW YORK • PARIS •
SAN FRANCISCO • WROCLAW

NEXT MONTH

Issue 26 on sale Thursday 28 March

>> LTSP

Discover the joys of diskless computing, and why a thin-terminal setup can save you time, money, effort and a lot of headaches. We look at the LTSP project and what it could do for you.

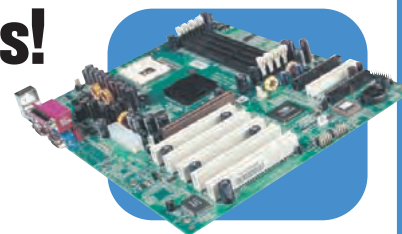
>> Even more tutorials

There's no let-up on the tutorial front – we'll have all your regular favourites plus Virtual Private Networking, Setting up a CVS server, using *Kpresenter* and the first part of a new series on *Apache*, by everyone's favourite networking guru, Dr Chris Brown.



>> And some hot reviews!

Including latest Caldera and Montavista distributions, Omnis *Studio*, *Revolution*, VIA chipset motherboards and more...



DON'T MISS YOUR COPY OF LINUXformat!

Get it delivered to your door every month – subscribe on page 96

The exact contents of future issues are subject to change