

Doing things better

When you think about why you read a particular magazine, there may be lots of individual reasons. When it comes to magazines like this one, most of you want to be entertained, informed and educated. From the response we get from the magazine, it seems that learning new things, or new ways to do old things, is very important to you. It's hardly surprising, as even a short tip can save you time, money and effort.

That's why we take our tutorials seriously and try to provide useful tips and advice at all levels. We want to help you 'do things better,' and in the meantime, you can help us do things better too – send an email, or post a comment on our website forums, and let us know what subjects you'd like to see tutorials on.

This month we have plenty for everyone. As well as your special Christmas gift of Mandrake Linux 9.0, we've tried to pack as much as possible into this issue. The bold assertion that

you will be able to build better code is backed up not only by a whole raft of tutorials, that stretch from compiling simple C code to the fundamentals of systems programming, but also extends to the biggest roundup ever of C++ IDEs.

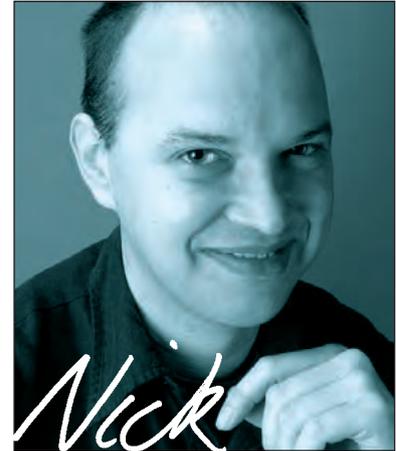
If you are more interested in using Linux than writing new software, check out the best new software in Hot Picks, or find out how to get more out of OpenOffice.org with our continuing tutorial series, which this month focuses on the spreadsheet application. There's also a special shopping feature for those looking for something a little more geeky this Christmastime!

The end of the year is also when voting begins for the *Linux Format* Awards. Last year thousands of you registered your votes, and made some developers and organisations very happy! This month we present the shortlists in each of the categories and invite you to make your mark electronically on our website – and may the most worthy win!

Our development coverage starts with a huge roundup of C++ IDEs p36

Stuck for gift ideas? Check out our gadgets special p50

You have the power to make someone happy! Peruse the shortlists and exercise your democratic rights. p54



Nick Veitch EDITOR

LINUX FORMAT

Aims of the magazine

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

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

Meet *Linux Format's* team of writers...



Richard Smedley
In spite of the obvious benefits to everyone, Rich still refuses to take part in our 'human modding' experiments

Maurice Kelly
Busy coder, electronic engineer and Midnight Oil fan.



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

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



Richard Drummond
With more time on his hands, Rich is no doubt developing a Java app to write his reviews.

Jono Bacon
Jono is a core KDE developer, web developer and writer. Jono is also a musician and sound engineer.



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

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



Charlie Stross
Master of Perl, Charlie has been writing about Linux for more years than anyone can remember.

Paul Hudson
PHP veteran Paul likes nothing better than implementing everything as a script.

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 p114

Newsdesk

Kernel developers banned from Free Software work; GNU desktop solutions; Budget Linux PCs; The Law turns to Linux; FOSDEM 2003; Passport for Linux; Crash simulation; Climate change predictions.

“SPIRIT OF THE WHIP HAND”

Bitkeeper controversy

The problems of using proprietary software on open source projects has been highlighted in Linus Torvald's decision to use BitMover's *BitKeeper* to manage development of the Linux kernel. It was a controversial decision to begin with, but a recent reading of the *BitKeeper* licence has caused leading lights in the Linux community to question the wisdom of using 'non-free' software on such a high profile project. The arguments on the kernel developers mailing list escalated until list admin Christoph Hellwig suggested blacklisting a critical Richard Stallman.

Kick off

BitKeeper is used to manage the various dev efforts involved in building the next gen Linux kernel and though it is proprietary, a cost-free edition is

available. The problem has arisen due to a clause in the licence for the free version which says: 'Notwithstanding any other terms in this License, this License is not available to You if You and/or your employer develop, produce, sell, and/or resell a product which contains substantially similar capabilities of the *BitKeeper* Software, or, in the reasonable opinion of BitMover, competes with the *BitKeeper* Software.'

In essence, people working on a project which may compete with *BitKeeper* are prohibited from using the free version. One such person was Ben Collins who, aside from working on Firewire support in the kernel, is involved in the development of *Subversion*, a replacement for CVS and, potentially, an open source competitor for BitMover. Collins told *Linux Format* he thought it was reasonable for a company to



require competitors to buy a licence, the problem he says, is that the language used in the licence is too broad (something BitMover says will be addressed in the next iteration).

"BitMover doesn't want competition taking advantage of the gratis license to use *BitKeeper* in order to further a competitive product.

"Larry (McVoy, BitMover founder) has given several roundabout reasons, but I think it boils down to the fact if one uses *BitKeeper*, one can deduce mechanisms to help add *BitKeeper*-like features to other programs. Since all of *BitKeeper*'s metadata is stored in the repository in plain SCCS format, it's easy to deduce the repo format. It's just a matter of deduction for the rest"

This, Collins says, would break *BitKeeper*'s unique selling point and have a real impact on the product's future.

"Their stance is, if you want to use *BitKeeper* and copy its features, you better be prepared to pay a license fee. There's nothing wrong with that. The problem is that their language is

very broad in order to make sure they catch all instances. This means some people may be inadvertently affected"

Ironically, the loss of *BitKeeper* hasn't affected the development of *Subversion*. "It was never meant to replace *BitKeeper*," Collins said. "Its current model doesn't follow the same approach that *BitKeeper* took. *Subversion* was always meant to initially compete with CVS, and do all the things CVS does, but do it better."

This doesn't mean *BitKeeper*-like features couldn't be implemented in the future. "It is several people's intention to add the features that make *BitKeeper* appealing to Linus and other kernel developers. Those features have been considered throughout *Subversion*'s development, so it's not as if it will be a major rewrite of the code to support it"

Not everyone has been as sanguine as Ben Collins though. Leading the anti-BitMover charge was Richard Stallman who said the restrictions on use were an outrage.

"Non-free software licenses are designed to divide and dominate the

```

ChangeSet 1.1001 01/01/29 15:18:51 InRurk@bitmover.com
El Patch: Grande mono->flags.
This is a big change to clean up various problems we had with
the flags as exposed by old trees (libbench) and logging trees.
It has the nice effect of making Bit generate pure SCCS files when
run in SCCS mode. For whatever that is worth.
src/bit-1.251-01/01/29 15:13:43 InRurk@bitmover.com
Add flags command to see the flags.
All fctool to take a list.

src/bit-01.250 (01/01/29 16:13:06) Diff 3 of 6 src/bit-01.251 (01/01/29 15:13:49)
-----
else cat /tmp/sync3#
echo == 42 is a superset of #1
echo ""
fi
/bin/rm -f /tmp/sync1231#

* For each file which is modified,
* call fctool on the latest vs the checked out file.
* If they create the merge file.
* then save the edited file in file"
* put the merge on top of the edited file
* fi
fctool() {
    cd $root
    case $(TMP)/fctool
    merge=$(TMP)/merge#
    prev=$(TMP)/prev1out#
    by=$(TMP)/by#
    test -e $fix || {
        echo Nothing to fix
        rm -f $fix
        exit 0
    }
    # XXX - this does not work if the filenames have
    for x in "cat $file"
    do
        by=$(TMP)/$x | #DPRDP
        echo #! $fix $x02 ytes $x04 $x04 $x04
    done
}

* For each file which is modified.
* call fctool on the latest vs the checked out file.
* If they create the merge file.
* then save the edited file in file"
* put the merge on top of the edited file
* fi
fctool() {
    if [ "$?" = X ] then _cd roots fi
    cd $(TMP)/fctool
    merge=$(TMP)/merge#
    prev=$(TMP)/prev1out#
    by=$(TMP)/by#
    test -e $fix || {
        echo Nothing to fix
        rm -f $fix
        exit 0
    }
    # XXX - this does not work if the filenames have
    for x in "cat $file"
    do
        echo ""
        by=$(TMP)/$x | #DPRDP
        echo #! $fix $x02 ytes $x04 $x04 $x04 $x04
    done
}

```

Keeping track of kernel development – is it worth losing freedom over?

CENTRALISED ADMIN

Red Hat acquisition eases network admin

Red Hat have purchased NOCpulse, the developer of *Command Station*, a remote monitoring solution which allows centralised admin of Linux, Solaris, Unix and Windows boxes. More significantly,

the software allows the monitoring of a cluster of computers as a single entity rather than a collection of individual nodes. RH plans to integrate the software into the Red Hat network.

Kernel news

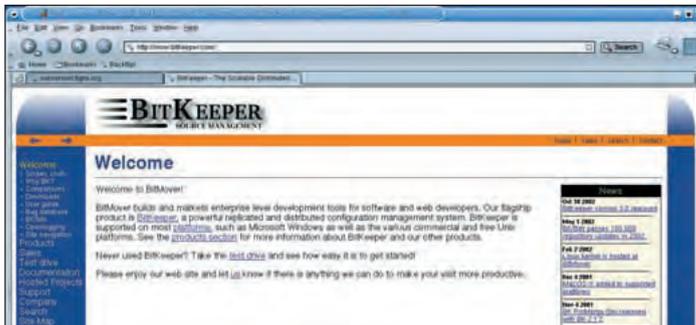
Roadmap for next stable kernel

The next version of the Linux kernel is planned for July 2003, Linus Torvalds told an assembly of hackers on the Geek Cruise in the Caribbean. After the much delayed 2.4 series, the core development team have taken a more structured approach to the next release. The 2.5 development release has now hit the 'feature freeze' point with the code freeze/bug squashing expected to begin early next year.

Some of the notable additions to the next version of the kernel include a major overhaul to Block Device management – the method used for piping data to and from storage devices such as CDROM, Flash cards and hard drives – and also the introduction for the first time of an API for the 'performance of cryptographic functions within the kernel.'

Find more at www.kernel.org/index.php.

The next version, by the way, will apparently be called "2.6" and not "3".



Not all Free Software developers are welcome to use Bitkeeper.

users, denying them the basic freedoms for software users. That's what makes them non-free, and that is what makes it wrong. Non-free software is a social problem, one that we need to solve if computer users are to have freedom.

"The spirit of the *BitKeeper* license is the spirit of the whip hand. It is the spirit that says, 'You have no right to use *BitKeeper*, only temporary privileges that we can revoke. Be grateful that we allow you to use *BitKeeper*. Be grateful, and don't do anything we dislike, or we may revoke those privileges."

This outburst brought a swift rebuke from McVoy who said his company wouldn't grant free licences for competitors. "The software is not open source because the open source business model doesn't have a prayer of supporting the development costs," he said. "If you

had built a decent system instead of sitting around and whining, we could be doing something else instead of sitting around listening to your whining!"

Hellwig began soliciting opinions on blacklisting Stallman from the list. Legendary kernel hacker Alan Cox countered that perhaps if RMS was blacklisted, they should also consider blacklisting Hellwig on the grounds of his "dangerous naivety."

So who's to blame for the whole debacle? Ben Collins says Torvalds picked the best tool for the job at the time, and the free software community didn't see the red flags and start working on something for him. "We couldn't have expected Linus to create his own tool. He's way too busy, and the last thing we need is Linus cracking and throwing in the towel!"

NEWSBYTES

■ At the end of October, **Abiword's** Dom Lachowicz announced on the project's mailing list that someone had raided *Abiword's* PayPal account to the tune of \$600. The thief then used his or her ill-gotten gains to buy a second hand camera, and also tried to extract \$1,200 from Lachowicz's credit card account. PayPal – after four weeks – finally looked into the situation and refunded the stolen loot. Lachowicz wrote: "To all of those who have written letters of support to both me and PayPal on my behalf, I thank you. I think that if nothing else, we've helped raise some awareness in the general community. At the very least, we've gotten our money back."



■ The government of **Spain's Extremadura region** has begun the process of migrating all public computer systems onto Linux. The final tally of desktops converted to open source software is expected to be in the region of 100,000. The project is the brainchild of the region's Education Minister Luis Millan Vazquez de Miguel who told *The Washington Post* that if Microsoft doesn't become more open and generous with its code, people will stop using it and it will disappear. "We are the future," he said. In addition to the 150,000 CDs of the distro being given to schools in the region, the project's website has had in excess of 55,000 downloads.

■ On a similar tack the **European Union** has committed €250,000 to a study to look at the viability of migrating member state computer systems to Linux. UK firm Netproject (see the *West Yorks Cops Linux* story) will test their system in the German state of Mecklenburg-Pomerania.

■ After a fairly long wait, Ian Clarke and his fellow developers have released a new version of **Freenet**, the large-scale peer-to-peer network created to bring censor-free publishing to the masses. Freenet 0.5.0.4 resolves some load balancing issues.

■ **Epson** have reinstated their – now GPL-friendly – printer/scanner software after withdrawing on recommendation from the Free Software Foundation. The problem, it transpires, was that part of the *libintl* package was used for internationalisation purposes, which contravened the GPL. As part of the corrective action, Epson changed their license to permit reverse engineering of the closed components, as specified in the LGPL.

Jono Bacon

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



COMMENT

Mandating freedom

“Freedom is an important word. It is the word that many take for granted in these times of unrest across the world. Although freedom means many things to many people, its concept is one that should be celebrated.

The idea that an OS can be developed around the world by volunteers which may include yourself is one that I personally found very exciting. It is this freedom that is celebrated in *Open Source* – the code is yours and you can see and do what you like with it; a concept born in a technical culture that has grown into other cultures where business and government take note.

Where this is getting interesting is that many groups of people are saying that it is a human right to have the ability to look at how something works – particularly at government level where your information is trusted to possibly closed sourced systems. Trust me folks ... I am no hippy – the hair and beard are due to my love of Heavy Metal, but my beliefs in our freedom stands true.

The *Open Source* effort and products such as Linux have taken basic beliefs in human nature and put them on the international social stage where governments and organisations are taking back the control of their info, and ours.

Radical as it may seem, many of our materialistic urges that brought us to Linux are facilitating a high brow moral side effect which has benefits for everyone. The Internet has been seen as a major revolution in human rights and access to information, is *Open Source* and Linux the next?

Stay tuned folks...it's getting exciting...

FREEDOM IN THE OFFICE

Desktop: The new battleground?

After the distro deluge last month, with their overarching lurch into the desktop space, that sector is now starting to look a little overcrowded.

As well as the resurrection of Corel Linux (under the guise of Xandros) and an updated Lycoris Desktop/LX joining the ranks of options on Wal-mart's range of PCs, Lindows have announced the first generally available edition (version 3) of their product. Moreover, Debian have leapt on the bandwagon with an initiative to bring 'really free' software to potential desktop switchers. The Debian Desktop project, launched by Chris Walters received an early boost with the opening up of Progeny's PGI (affectionately known as Piggy) installation routine. Walters said the

motto of the project should be 'Software which just works', and this idea should be evident in every stage of use, from installation to software updates.

Meanwhile SuSE have announced an Office Enterprise Desktop Edition (OEDE) of their latest distro designed with disgruntled Windows users in mind. Like Xandros, OEDE ships with Codeweaver's excellent *Crossover Office*, meaning enterprises can upgrade without alienating staff trained on MS *Office*. The distro also contains a full version of *StarOffice*, so MS-free environments can access attachments in the ubiquitous *Word* format. Other significant additions in OEDE include a *YaST* update which examines an existing Windows setup before suggested a suitable 'reorganisation' of the hardware space,



A Debian subproject aims to bring the joys of apt to the desktop user.

and a new partitioning tool from Acronis.

Finally, the inaugural Desktop Linux Summit is scheduled to take place in San Diego on February 20, 2003. The event has already attracted the interest

of the likes of Hewlett Packard, Mandrake, Lindows, Codeweavers and Ximian. Find out more – and even register – at: www.desktoplinux.com/summit

Linux Web Watch/



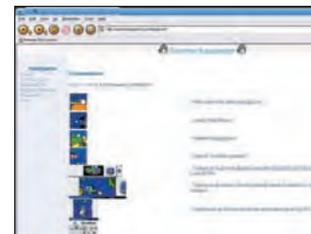
Flash – 'savers with content.



Big Ben – time to waste?



Amor – does what it says on the tin.



Aquarium – almost useful.

Almost pointless?

There are some things available for your computer that really don't do much of value, but are interesting nonetheless.

Back in the day, Screensavers were an essential tool to stop long displayed images being 'burnt' into the screen, leaving an annoying ghostly shadow. Screensavers combat this by kicking in after an specified time with moving graphics, which obviously cause the screen to update. A quick trip to www.swift-tools.com/Flash provides an RPM of a tool which will give you the option of using any Flash movie as a screensaver. It's KDE only at present.

Big Ben (<http://ethereal.net/~wampa/>) is a small clock applet which adds an unspeakably bright range of clocks to your desktop. Its author says that though the clock is a small program now, he plans to 'bog it down with more features than is considered sane.' First up is a talking feature. Great!

The *Amusing Misuse of Resources* (*AMOR*) could be the most pointless app we've ever seen, but it is

themeable. So that's alright then. *AMOR* essentially puts an annoying animated avatar on your desktop which, er, does nothing but distract your attention from doing anything useful. The website tells you helpfully how many users have been disappointed by the page. Get it, or don't, at <http://homepage.powerup.com.au/~mjones/amor/index.html>. Sherman's Aquarium ([\[aquariumapplet.sourceforge.net/\]\(http://aquariumapplet.sourceforge.net/\)\) is an *Xsaver* hack that puts a humorous \(and mildly useful\) aquarium on your desktop. It can also be docked as a *Window Maker* or *GNOME* applet. The usefulness comes from the fact that the aquarium can give you an idea of CPU load, time or the status of the **Num/Caps/Scroll Lock** keys. The fish are based on Jim Toomey's *Sherman's Lagoon* comic strip \(\[www.slagoon.com\]\(http://www.slagoon.com\)\).](http://</p>
</div>
<div data-bbox=)

NEWSBYTES

■ Another month, another **Phoenix** release. 0.4 of the stripped *Mozilla* browser features Typeahead; Find; Theme support and an even smaller footprint. At this rate, the project should be at zero k by about v1.4.

■ Patents No. 5,576,951 and No. 6,289,319 appear to give **PanIP** the sole and exclusive rights to the idea of e-commerce. Since April the company have launched lawsuits against numerous small web business hoping to build a critical mass of precedents before taking on the likes of Amazon and Wal-mart. Initially PanIP was said to be asking for a 'one-off licence fee' of \$5,000.

■ Norway's **KISS Technology** have joined forces with DivXNetworks to bring DivX playback to a consumer-focused DVD player for the first time. The DP-450 should be in selected outlets as you read this. It is capable of reading DivX 4.xx and 5.xx and should cost somewhere near 400.

■ A study has suggested that many **US and European banks** are looking to Linux to reduce costs following the recent 'upgrade' to Microsoft's licensing regime. One corporate buyer also noticed a problem with the MS automatic update service, the licence for which gives MS unlimited access to your systems – a data protection no-no.

■ Cutting edge **XFree86** users should now be able to resize, flip, rotate and change refresh rates on the fly thanks to recently added RandR support. www.xfree86.org/~keithp/talks/randr/randr.

■ **XML** continues its march to ubiquity as MS announce the next version of *Office* will have file formats based on the language. Tim Bray, who was involved in the creation of XML, said that *Word* and *Excel* (and of course the new *XDocs*) can export their data as XML without information loss. "Built around an open, internationalized file format, *Office 11* is going to be a huge step forward for management, independent software developers, and MS," he said. The XML 1.1 spec was released, but the decision to break backwards compatibility – changes to the spec's treatment of Unicode – with version 1, led some commentators to suggest the World Wide Web Consortium was favouring IBM.

■ **LG Electronics** in India have decided to begin marketing a range of low-cost Linux computers. Called My-PC, the range is expected to cost 8-10% less than foreign brands.

■ After the slightly extended beta period, **United Linux** are ready to ship a final version of their base distro.

■ UK ISPs have balked at keeping reams and reams of personal information on users, rejecting a central tenet of the government's **Anti-Terrorism Crime and Security Act** passed a year ago.

CHEAP LINUX PRE-LOAD

Evesham target Lindows switchers

Evesham Technologies have become one of the first major vendors in Europe to begin shipping a Lindows-based product. The company's E-Scape (geddit?) line starts at just £249.99 (without monitor) which is broadly in line with Wal-mart's US offerings. Add in a basic 17" screen and the price leaps to a paltry £369.88.

Built around a Via Eden C3 800MHz processor, the E-Scape comes with 256MB RAM, 40GB hard drive and CD drive. Net access is taken care of by an integrated 56kbaud modem. Evesham



Model shown with 15" LG 1510 TFT

Evesham usher in a new generation of desktop machines based on LindowsOS.

are pushing the machine as a low cost, basic machine, ideal for web browsing, emailing and 'general office work'.

DEVELOPER CONFERENCE

FOSDEM 2003

Brussels will once again play host to the Free and Open Source Software Developers' European Meeting (FOSDEM) on February 8-9. The two day event will be attended by some of Free Software's leading advocates and developers for a series of 'tightly focused developer to developer' seminars on a range of subjects.

The event has grown steadily from its debut in 2001, when 1000 developers made the trip to Brussels. Last year the event attracted 1,600 attendees and Richard Stallman presented Python developer Guide von Rossum with an award to acknowledge his contribution to free software. RMS will again play

the rôle of compère in the 2003 Free Software Awards at the event. Speakers include Jon 'Maddog' Hall, Owen Taylor from *GTK+*, Julian Smart from *wxWindows*, Havoc Pennington on Free Standards, David Faure for KDE, Michael Meeks for GNOME, Bruce Momjian from PostgreSQL, and many others.

Organiser Damien Sandras said "FOSDEM's goal is to provide Free and Open Source developers and communities a place to go over the latest developments in the Free and Open Source arena and to promote the development and the benefits of Free and Open Source solutions."

www.fosdem.org.

FAST TERABYTE SORTING

IBM's supercomputer file system comes to Linux

IBM's General Parallel File System (GPFS), optimised for video and audio production, features cluster management and collaborative tools geared toward digital content, enabling animators, sfx artists, editors and directors to work on the same project regardless of their physical location.

GPFS, a core component of the Accelerated Strategic Computing Initiative (ASCI) Blue Supercomputer, allows parallel programs running on up to 488 nodes to read and write files at up to 1.2GB/second to file systems in excess of 15TB.



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

Un-happy Old Year

“ Out with the old. In with anything else. It's been a long year with nothing really worth cheering about.

The smart money is attempting to pimp a drop-in desktop replacement for Microsoft Windows. Appealing to the IT manager (It's easy to manage!) and the CFO (It's cheap!), some may even have the poor user in mind as they all attempt to out-Window Microsoft.

Haven't we seen Linux succumb to the same Balkanisation that throttled the commercial Unix market? The only strategy that makes sense – the Linux Standards Base – is still so far below the radar as to be generally ignored. Anyway, putting incompatible things in the same place seems to be of no real help.

I still don't have a decent 3D accelerated driver for an S3 video chip for my laptop, yet the coders at XFree86 are working on a Rotate and Render extension so I can turn my display sideways. Very cool; let's hope the Render part is useful.

Even with improvements to the insanity that is application packaging, it's still amusing to watch some poor newbie install them on one of the Win-clone distros, or even attempt to install that distro on their Compaq/Dell/HP beast from last Christmas. What fun!

At least Microsoft is making the New Year cheerful with their arrogant licensing scheme that is destined to drive every thinking world government to Open Source (meaning the US is sticking with the Redmond Gang on this one). Will the New Year be better?

We can only hope so. ”

DESKTOP ROLLOUT

West Yorks cops Linux

West Yorkshire's police force have become the first force in the UK to see GNU/Linux become their standard desktop OS, following last year's study by the Police IT Organisation.

The West Yorkshire rollout covers approximately 3,500 machines but if the trial is successful, the entire police service for England and Wales (some 60,000 desktops) could be converted to open source software.

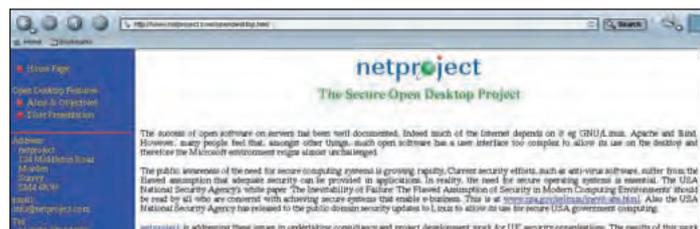
The VNC-based system uses a 'stateless' system so users can log

onto any machine with their SmartCard and have immediate access to their own suite of applications which, by default, will include the latest version of *OpenOffice.org*, and allows access to legacy systems.

The SmartCard reader-equipped terminals are low-cost desktop PCs from CGI.

Netproject, which won the contract to design and maintain the system claims the whole project will save up to £1 million per year.

www.netproject.com



Netproject's Secure Open Desktop Project could form the basis of a national police open source platform.

PLATFORM SERVICES

Progeny look beyond distro

Following the demise of Progeny (the easy-to-use Debian distro) Ian Murdock is back with a new plan to bring Linux to the corporate masses, and this time he hopes to deliver 'Linux' rather than Red Hat, Debian or any of the other flavours that he says can lead to confusion.

Murdock told Linux Planet that his experience with Progeny demonstrated that many potential users – especially in the embedded space – were interested in 'pure Linux' without the kitchen sink method of packaging that goes into most distributions.

"We saw an opportunity," Murdock said, "We manage the underlying platform and integrate our customers' technology. This allows our customers to focus on the value-added functionality that differentiates their offerings in the marketplace."

What this appears to mean, beneath all the spin, is that instead of offering a standard solution, Progeny Platform Services can be



From the ashes of the distribution business model, Progeny rises with a brand new plan.

hired to design and build the perfect Linux, tailored to the customer's requirements – which should decrease time-to-market, and save the cost of a lot of in-house development.

The company's first big name

contract was with HP who hired Progeny to create a Linux development environment to speed up work on the company's Linux applications.

www.progeny.com

SIGN ON SERVICES

Ready-to-Run prep Passport for Linux

Software developer Ready-to-Run have apparently been granted a licence to bring Microsoft's Passport authentication service to Linux and Unix. The original Passport Manager software was designed to be exclusive to Windows technology, but after pressure from large Passport customers MS relented and contracted Ready-to-Run – who had already done much of the work to offer a Unix Passport service to the Pressplay music service – to prep a *nix solution for general release. Their first release will offer support for Solaris with Red Hat, AIX and HP-UX following.

Ready-to-Run are thought to be the first in a line of developers who will bring the service to various OSs under the MS Shared Source initiative which gives major developers limited access to MS source code. Ready-to-Run have had access to the .NET code for the past two years and have so built and deployed sign on systems on numerous platforms.



It takes three racks of these to crash your car...

LINUX CLUSTERS

When you want your computer to crash

DaimlerCrysler have become the latest in a long line of enterprises to discover the cost benefits of Linux and open source software. The car giant has recently purchased over 100 IBM dual processor Linux machines to run their crash modelling system. IBM will

also provide storage solutions and manage the cluster. The move is seen by Big Blue as another coup for their Intellistation range after high-profile orders from the likes of WETA digital and a number of academic institutions. DaimlerCrysler had previously relied

on a proprietary Unix cluster, but have looked to Linux to reduce costs. The system is built on 108 IBM IntelliStation M Pro 6850 workstations, each with two 2.2GHz processors, 1Gbps NICs and an IBM TotalStorage FastT500 storage system with 2.6 terabytes of capacity.

CLIMATE PREDICTIONS

Clusters designed to save the planet?

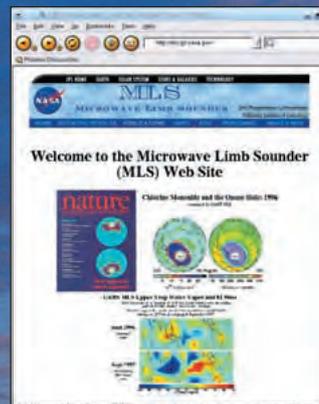
NASA's Jet Propulsion Lab (JPL) have tapped into the USA's TeraGrid an uber-cluster of 1A-64 processors in an attempt to improve the accuracy of today's climate models and predict the potential damage caused by climate change over the next couple of decades. And if you imagine how the potential for inaccuracy in a five-day weather forecast, you'll have some idea of how big a job this is.

The TeraGrid, based at Argonne National Laboratory, Caltech, the National Center for Supercomputing Applications and the San Diego Supercomputer Center, is linked via

30-40 Gbps connections and has a combined computing capacity of 15 teraflops. The project's storage capacity tops out at more than seven petabytes. JPL's own cluster is no slouch either, consisting of 25 networked Pentium III PCs each with 2GB RAM.

Dr Jonathan Jiang, a scientist at JPL's Microwave Limb Sounder team, told Earthweb that creating an observational model could take 10 or 20 years. "By that point it might be too late for us to prevent the climate changes," he said.

www.jpl.nasa.gov
www.teragrid.org



The Microwave Limb Sounder team will have 15 teraflops of power at their fingertips.

IMAGE MANIPULATION

HTTP image editing

TrueSpectra are bringing their respected HTTP-based image manipulation server to Linux for the first time. *Image Server* integrates into any web development environment (or as a plugin for *Apache* or *IIS*) and allows the developer to manipulate images on the fly using standard HTTP query strings. CEO David Watkins said the company had seen 'significant adoption across a range of sectors' within the Web industry.

"We've continued to listen to developers and have built upon our ease of use in 4.1 with enhanced

installer, design tools and broader OS support. The overwhelming response by customers convinced us that v4.1 will extend our lead in the market."

Doug Dawirs, Director of Online Services at Workbook & Company (www.workbookstock.com) said the server dropped into their operation with very little fuss. "The transition to TrueSpectra Image URLs in place of static image tags was easy, and the resulting simplification our workflow has already saved us money and effort"

A trial version is available from:
www.truespectra.com

POWERFUL PIM

Groupware everywhere

Inspired (perhaps) by the KDE

Kroupware project which has garnered official backing from the German government (*LXF 34*), SuSE launched their own project targeting the backend business. Their *Exchange*-alike product, *Openexchange Server* gives users access to email, calendar and PIM functions through web browsers. *Openexchange* offers claimed TCO savings of 59-64%. It is also the most powerful Lotus *Notes* solution on commodity x86 hardware, capable of 6000 users per server.

David Cartwright

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



COMMENT

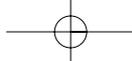
How green is your grass?

“ Contributing recently to a Linux newsgroup, I anticipated starting a bit of a flame war. Someone was suggesting that it shouldn't be possible to persuade a Linux user that anything outside that world was worth using. I didn't mince my words in my response, saying anyone unwilling to consider anything but their "favourite" technology was at best stupid and at worst unemployable. I couldn't help wondering, as I hit "Send", what the tone of the average response might be. To my shock (and delight) the responses I've seen so far have been completely positive.

Of course, this doesn't just mean us Linux types have to occasionally consider an MS, Sun, Novell, Cisco or Nortel solution, but also that those misguided souls out there from other camps ought to peer over the fence into our Open Source garden every once in a while.

One must also remember, of course, that the best technology is not always the one to choose. As I said in the same newsgroup posting, there's a technology I'm looking at right now (from a large, well-known vendor) which is technologically the right tool for the job but whose salespeople are jerking me around and making me wonder whether my after-sales life will be hell if I do spend my money with them. But here again, it's just another situation where I'm peering over the fence to see if the other side's more attractive.

The moral of the story: take your blinkers off and consider options outside the Linux world. And, of course, point out to those on the other side of the fence that they might get a pleasant surprise if they peer over the fence into your garden. ”



Mailserver

Mailserver

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

★ Letter of the month

This month's winner receives a copy of Sun's StarOffice



The power of LUG

I've been using Linux as a desktop OS on and off for about a year and at last I feel that I might be getting somewhere. If anyone else out there feels like giving up after four months or so of dual booting then my advice is "don't! – the worst is past." I first started with a RH6.0 'publishers edition' picked-up from a bargain-bin bookstall, and immediately got into trouble with trying to configure MS Windows friendly devices (modem, printer) and not knowing anything about how Unix operated or how to install anything that wasn't on my original (limited) install disc.

I bought a copy of *LXF*, and I have to admit found about 75% of it totally baffling – but it did put me onto my local LUG (Anglia), and that saved me. Just lurking was enough to see that these people were friendly and knowledgeable. In the

end I plucked up courage to ask some (pretty dumb) questions and got useful and helpful replies. Now eight months on I have a dual-boot Win98/RH7.2 box with KDE 3.0 (courtesy of *LXF*), a 386 running 'TinyLinux' and I have Slackware 8.1 (again courtesy *LXF*) on a 486 with 16MB Ram that does pretty much everything I want from a computer.

The diversity and philosophy of GNU/Linux is so far removed from the uniformity and dogma of MS Windows that I can now see a future without MS, but not without the support of a Linux User Group. My advice to all newbies is join-up to your local LUG, ask and you shall receive. Thanks to *LXF* I also have *OpenOffice.org* installed on my RH7.2, which I'm really pleased with, but why doesn't OOo support import of *Gnumeric* and *Abiword* files (which I use with Slackware)? As far as *LXF* itself, I'd like more docs on the CD

occasionally (I know that the DVD has the LDP files on it regularly), and a piece on scientific software (there's loads available for Linux – from GIS, Biological Modelling, Statistical analysis, etc.) would be welcome. Also I must admit that I'm still pretty much in the dark about OpenGL which seems vital to gaming on Linux, how about covering it in the mag and putting drivers on the disc?

Glen Tyler, *via email*

Never underestimate the usefulness of your local LUG. We did do a short series on OpenGL programming (which, by the way, is only useful for 3D applications) a while back. Until something similar comes around again, you can find loads of useful info at <http://nehe.gamedev.net/>.

For your sterling work in LUG advocacy, we're awarding this month's star prize, a copy of Sun's *StarOffice*!

a third of my monthly income...

You can't fool me, this one's for the business user as well, isn't it? Is there any chance of an icon to show if a review is 'One size fits all', or if it's intended for those-guys-with-all-the-money or their victims? And in the meantime can you recommend me a firewall, soft or hard, that will work, won't wipe my disk, break the bank, or fill my screen with 'nix code? Deke Roberts, *via email*

SmoothWall is a firewall system intended to turn an otherwise redundant system (i.e. an old PC) into a useful device again – a firewall. As it needs to install its own distro, and to be of any use needs to be running all the time you are using the computer, there is no great point making it capable of dual booting. Many firewall distros follow the same philosophy, because it's easier to make a system more rugged this way.

A standalone firewall device is usually more expensive because, er, it's a piece of hardware. In fact it is a computer, just one that has been optimised for performing a specific task. This is not a review for 'big business'; it's just a review of a standalone firewall. That's how much they cost. Your computer cost more than that.

Anyway, you don't need to buy anything. Firewall capability is built in to your kernel. All you need to do is configure it. Happily there are some tools to help you do this. I would suggest taking a look at *FirewallBuilder* for example (www.fwbuilder.org).

Archives

Firstly I want to say how much I like *LXF*. I live in Germany and although my German is not at all bad, I really appreciate it when I can read a Linux magazine in my own language!

As a new subscriber – I have been reading *Linux Format* off and on when I have been able to find a copy here – I find it a real shame that I am not able to read previous

Brick wall?

The trouble with using Linux on your home computer is you're a member of such a small community that it's not worth any major publisher's while to bring out a magazine for you, so you share it with big business.

And so it was, that a couple of weeks ago I wrote and asked you to put Smoothwall on the DVD for me and a couple of days later it came through the door. 'What service!' I jested with myself, and almost loaded it straight on to my machine. Fortunately I read the instructions first. "Smoothwall reformats the entire drive, you'll lose everything

on there." Why? Why in the name of tarnation would a Firewall want to wipe my disc, and why didn't you mention this when you recommended the thing a couple of months ago? Quite suddenly candidate one for firewall duties on my newly upgraded-to-Linux PC has died, and I must needs find a replacement. I flip through the magazine, and THERE IT IS! The Snap Gear Pro+.

A hardware firewall. What a great idea! And it only costs HOW MUCH?!

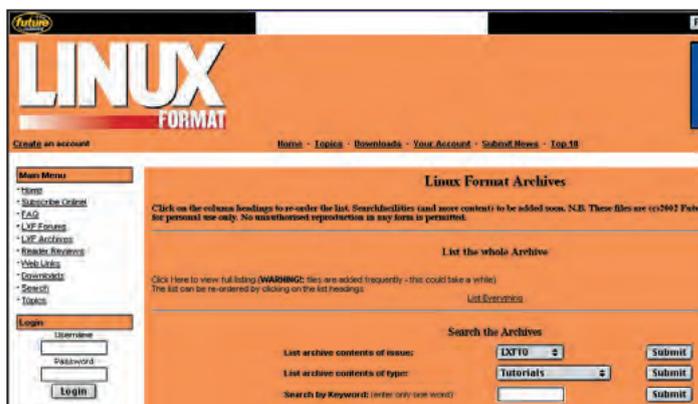
This'll be where my opening paragraph kicks in. This device is not for me. I can buy a computer for that outlay! So I go in search of a home version and there it is, just like you

said, the Snap Gear Lite+. I stifle my desire to wring its scrawny little neck for being called 'Lite', it's not its fault after all, and I read on. Oh much better, it's only 200 quid. Just under



SnapGear – a computer optimised to run a firewall.





Past articles make their way to the LXF Archives on the website.

tutorial installments. I have for instance been trying to follow the Perl tutorials, and the only way I could catch up on installments I missed would be to purchase the relevant back-issues. Now I know you would be very happy if I did this, and if that is the only way I can manage it I may very well just have to fork out for several back issues. However, it occurs to me that it would be a fairly simple matter to archive tutorials once in a while on the DVD.

If this is too much like giving away the family silver, then perhaps you'd consider archiving such articles for download by people who subscribe to the magazine. I subscribe to *New Scientist* and they allow subscribers access to the archives of every article for the last ten years. Now perhaps archiving all your articles would be a little excessive, but it would be a very useful service in my opinion.

Francis Pressland, *via email*

Thanks for your comments, I'm glad you are enjoying the magazine. Many of the articles are available online in PDF format. Just visit the *Linux Format* website (www.linuxformat.co.uk) and follow the link to the LXF Archives on the left hand side.

We could put these on disc too, but altogether they would take up too much space. It may be that we will include some of the more frequently requested items on DVD soon.

Installing

As a new Linux user, I have installed Mandrake 8.2 and am slowly getting to grips with the operating system after many years with Amigas and then PCs running various versions of Windows, I am getting on quite well.

However having purchased your magazine and tried to install a couple of games now, I have to say that

either I am "thicker" than I thought or you CDs leave a bit to be desired. I tried to install *FlightGear*, it didn't work, I also tried to install *Vega Strike* with no more success.

I am not alone in wanting to do more than just surf and write letters on my computer, and this sort of problem does not encourage other Windows users to try Linux. I really like the Mandrake system and it installed no problem, so it must be the game installers or lack thereof on your cover disk. Please either print proper instructions or have good installer software on your discs. I really want to use Linux, but I don't want a two day fight to install every piece of software that comes my way.

That aside I do enjoy the magazine, and have passed it on to a couple of Windows users that have spare PCs in the hope they will be tempted to try Linux as well.

Jim Robertson, *via email*

Sorry you seem to be having problems, but without specifics, it's difficult to help – though I would suggest you try the help forum on our website if you have problems in future. The *FlightGear* CD was optimised to be run from CD, not installed, although it is possible to do



FlightGear ran from the CD. We can't make installers for everything!

this (there are instructions on the CD). You should be able to run it directly by just typing in the command listed in the mag and on the CD.

We don't create the software on the CD, and apart from very occasional projects, like *FlightGear* where we set it up to run from disc, so I'm afraid we can't create installers for everything. Most source will compile fairly straightforwardly if you have the requisite libraries, etc.

OpenOffice.org

For two months I have been like a bear with a sore head. I had the Aug

not the mag, the *machine*, out of the window. I was not impressed. I could not get the damn thing installed.

I knew it was me. I am new to Linux. I run RH 7.3 and only maintain a separate hard drive with W98 'cos I have some legacy hardware that a lack of £££££ prohibits an upgrade. Imagine my dismay when, in your November mag you have a full four page spread on OOo. Damn. Looks so good and I cannot get it working

Still, I read it. What was this – an idiots guide to installation?

“You don't need to buy anything, firewall capability is built into your kernel all you have to do is configure it, and tools are available to help you”

LXF, DVD in hand. *Openoffice.org* poised to be installed. I had a grin the size of Lake Michigan. Three weeks later, I nearly threw the machine,

Within 25 mins, I had OOo installed and working and had created my first written document – woohoo! Hats off to Sun for making it so





Mailserv

« easy for even me to install. Hats off to *LXF* for putting it on the DVD and hats off again for making the 'install guide' such a good read and so easy to follow.

Keep up the good work *LXF*
Mark Daniels, *via email*

Glad you found the install guide useful – *OpenOffice.org* is a great piece of software and I hope you'll find the rest of the series will open up further delights for you.

Where?

I write this letter in response to *Why Windows?* (*LXF32*).

Congratulations mate, but I can't make the giant leap and reach the other side, even though I tried several times. I hereby voice my concerns on the compatibility that Linux needs to become a genuine force in the desktop market. This may very upset some hardcore Linux users since Linux did improve extensively in this area lately, but allow me wake you up a bit to the real world of the every day desktop user.

I am still a newbie, for some time now, since Red Hat 6.0 and *StarOffice 5.0*, and am still downloading and trying the latest distros, including Mandrake, SuSE, Lycoris, Blue, ELX, etc. Installation of Linux has largely been perfected, in my opinion, and hardware issues are seldom found.



Take advantage of our install guides and tutorials to ease your worries.

Compatibility with expensive office suites has been overcome in general as well, and we all know the exceptional server capabilities of the penguin. These were the main concerns for the Linux hierarchy in the past few years, but let me draw your attention to the future, especially since predictions are that governments will convert to Linux soon in the near future. Installing new software can be a real pain.

After giving up on .gz files (because compiling isn't something your normal enthusiast do), RPM's are also becoming increasingly dodgy.

Double click a rpm these days and one has to really search and dig deep, hoping to find your newly installed program somewhere. No shortcuts are provided on either the desktop or start menus, not even the option to do so. Then one day you log into GNOME, only to find it hidden in some sub sub directory. Do you now use GNOME or KDE? Not to mention the others like *Enlightenment* etc. That discussion is probably asking for trouble, so couldn't all desktops simply use or access one start menu folder for example? Some new distros (like Lycoris making network browsing

so easy – why don't the big distros use this easy network integration?!?) that I tried are KDE-based, and therefore some very useful software simply can't run on them.

How many users will convert if they knew *MSN Messenger*, *RealOne* or even *Netmeeting* could be used on Linux? Now I know some guru out there is probably saying that I can use all this software if I only... and that my friend, is my point. If I, who am keen to use Linux and want to see it succeed and to be used by everyone, have a difficult/impossible time getting things to work on my favourite distro, how will we convince other regular users to use it? Until this nightmare with different start menus and desktop dependencies has been solved, Linux will have a hard time trying to convert your average desktop user.

My solution is simple – make it simple. Simplify start menus, access to icons, network browsing with *Samba*, installing software on various desktops and making them work on all desktops. Even though the latest Linux distros look great, a shelled OS will remain my desktop of choice on my dual boot system at home, for the time being that is...

On the Expo, nice and friendly staff *LXF*, keep up the good work with your great mag!! Where were

Helpdex

shane_collinge@yahoo.com





Red Hat, Mandrake and the rest? Even Microsoft had a stand there. Laurence Munro, *via email*

I can see your point regarding installation, but I can also see that some people don't want software to automatically install itself in menus. In any case, this is quite possible to do, even with rpms. The main difficulty is that, as you mention, it may be difficult to determine where desktop shortcuts should go, etc.

However, it shouldn't be too hard to track down software you are actually installing, particularly if it is a desktop application that you are likely to add to the menu. Try

```
whereis <name>
```

from your shell to locate an executable.

Save our CD

A quickie in response to your request for comments on the *Save our CD* letter:

I'm very glad to hear you have no plans to ditch the CD (*NB* I have

“They are excited about not having to pay obscene licensing costs and even more excited about the prospect of enjoying systems that run reliably”

just upgraded my subscription from CD to DVD).

Apart from the apps (and the source! Source code can be a resource in itself quite apart from the app it is part of) the essentials (on the DVD at least) are really great to have by my desk. It is “very” good to know I am now always up-to-date with the LDP. Also, some of your readers may not be aware that the LDP mirror you provide contains two monthly magazines – the *Linux Gazette* and *LinuxFocus* (which itself is in a ton of different languages).

Finally, I have the impression that you would/do listen sympathetically to readers' requests for software on the CD/DVD.

The CD/DVD is definitely an integral part of the magazine as you say (found it very useful with last month's emulators article). It also strengthens *LXF's* position as a community magazine.

Ivan Uemlianin, *via email*



Linux – winning over business users with common sense arguments.

As we said, there are no plans to drop the CD or DVD. Indeed, as you've noticed this issue, we've even given you an extra one!

Success

Having been to the Linux Expo in London yesterday, I first wanted to say that it was certainly a nice touch to have been given a *Linux Format Magazine* for free; when I saw all the software that is included

on the DVD, I appreciated that even more. Now for my Linux success story: I had been working for a small Media company in Los Angeles. This company has, while I have been finishing my Bachelors degree in Computer Science in Southampton, been steadily growing, and, more importantly, been shifting its business core away from text-documents towards multi-media handling and Audio and Video post-production.

This means that whether they like it or not, Information Technology and Strategy are playing an ever-increasing part in their business model. Having used Linux since 1997, I supplied them with two Linux machines which had SuSE 7.2 installed. One of those machines was a dual-boot system with Windows 2000, and the other one pure Linux. But now, that I've finished the Computer Studies degree, it was time to make a serious contribution and change in direction for them. Two

friends from University and myself are finishing a J2EE-based in-house work-tracking and billing system for this company. It will run on an *Oracle 9i db*, running on *SuSE 8.0*, in conjunction with a *JBoss 3.0* middleware server, running on *SuSE 8.0* as well. They are excited about the prospect of not having to pay obscene licensing costs, (an important factor for a small company), and they are even more excited about the prospect of enjoying systems that run reliably. The company and I have also been planning for several other important additions to their IT infrastructure: a *Smoothwall* firewall, a dedicated *SSH* server for quick file down-and upload for their clients, and a *Qmail* server to handle all company mail. The reason for choosing *SuSE 8.0* for these machines was simply that I am most familiar with this distro, and it has performed well for me; I have been quite happy with it. This case study proves that Linux is coming, and it is empowering small businesses to provide services to their clients that would (usually) be expected from companies bigger in size. Under my guidance, the percentage of Linux to Windows/MAC OS machines has changed from 0% to ~30%. And those are the 30% that count most, because they are/will be the “nuts and bolts” of the operation. I'd call that a success story!

Volkmar Woinke, *via email*

Thanks for the details of your work in progress. I agree that small businesses are big winners in a switch to Linux, but they do need to find the help of knowledgeable Linux people like yourself. [LXF](#)

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: lxf.letters@futurenet.co.uk



**Reliable, Fast,
Effective
Web Site Hosting**

£59

+Vat Per Year

Includes the following
(no hidden costs)

100MB WEB SPACE
BEYOND BROADBAND SPEED
99.99% UPTIME

Your Own Domain Name
www.yourname.com
(other domains also available)

10 x POP3 email accounts
with forwarding
and autoresponders

Windows and
Apple Mac Friendly

Macromedia FLASH Compatible

FrontPage 2002 Extensions

 PHP4 Scripting
Enabled
Example Form Processing Scripts

24Hr FTP Access
Online Control Panel
Graphical Site Visitor Statistics
Free Technical Support

No Banner Ads –
unless you put them there!
Your own virtual server
under your control

Quality UNIX Servers housed in
a State of the Art UK Datacentre

Other Options Available:
MySQL Database, SSH, Perl

Find Out More and
Order Online at...

www.hotfibre.net

 **hotfibre**[™]
web site hosting

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...

Mandrake 9

The *Linux Format* readers' favourite reaches another milestone release. Will it be taking you with it? **p19**

Libranet

Debian made easier for the desktop. Find out if this is the distro for you **p33**

NetOp

A remote admin tool that shines — but can it outshine Free Software remote admin tools? **p22**

VirtualPC >>

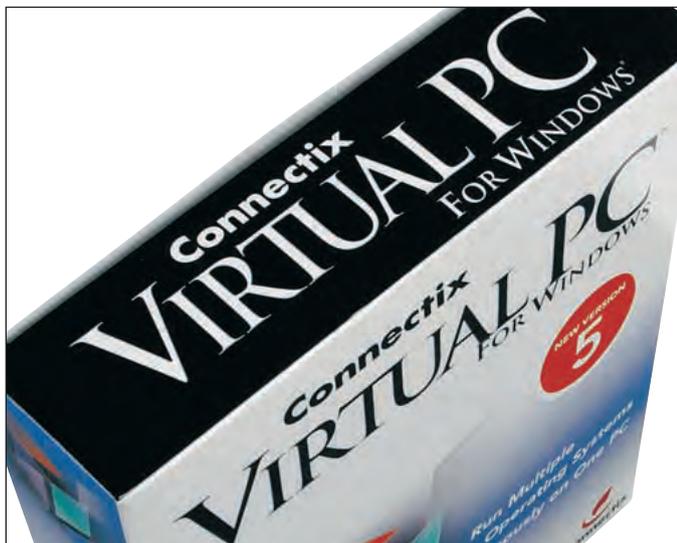
Linux everywhere — even on your MS Windows desktop **p26**

VueScan >>

Is *SANE* the be-all and end-all of Linux scanner software? Discover this powerful commercial rival **p28**

Books

Fill the long winter nights with *Practical Python*; *Writing Perl Modules for CPAN*; and *Perl in a Nutshell (2nd Edition)* **p34**



COMING UP SOON...

RealSoft 3D

3D modelling, animation and rendering. Finnish firm RealSoft port to Linux.

UT 2003

The Demo has reached our award nominations, but we're still awaiting the full release.

Xandros

Corel Linux comes back from the dead — as an easy-to-use desktop system.

Jool Kwartz

If you saw this server at the Linux Expo, you may be wondering if it's your SOHO solution.

Arkeia

Arkeia aim to make back-ups easier — as well as more secure. Read our review next month.

Webmin 1.0

The incredibly versatile, web-based remote admin tool reaches its milestone release.

DESKTOP DISTRO

Mandrake 9

What's cute and cuddly and at the cutting edge of technology? No, it's not Nick Veitch, but he reckons it might be the latest Mandrake release.

User-friendly, desktop-oriented distro – up against recent releases from Red Hat and SuSE.

- **DEVELOPERS** Mandrakesoft
- **PRICE** Free download, to €199 box set with manuals and support.
- **WEB** www.mandrakesoft.com

There's been a long history of Mandrake being a popular choice for desktop users, way before the likes of Red Hat got interested in the market. Mandrake also have a reputation for providing the very latest tools and applications, with varying degrees of success. This latest release probably isn't going to challenge any of your preconceptions.

Installing

Setup and discovery is good. The commercial versions include the nVIDIA drivers too, which the download editions do not because of licensing. However, the download edition also neglects to mention that you can download the drivers from nVIDIA if you wish, which might have been nice.

Other hardware is well supported. This is the only distro we have noticed that detects stuff like a graphics tablet during setup, and configures it properly. The wheel mouse

configuration utility still goes a bit berserk until you press the middle button, which might be confusing or frightening for some, but at least it works.

Hard drive preparation is second to none, thanks to the excellent *DiskDrake*. Comments have been made that it should be more obvious what's going on, but frankly, this is the best partitioning tool available to Linux, especially in an installer. XFS support is available, as is the option to encrypt filesystems, as well as support for reading NTFS partitions.

Package selection is always going to be contentious. The various software is grouped together for selection, but anyone who has used Linux before will want to at least look over the final package list to make sure their favourites are included.

For a more extensive test we also installed via ftp, first as a new install, and secondly as an upgrade to a Mandrake 8.2 system, which itself had been upgraded with various bits and pieces in the past. Both worked flawlessly, and surprisingly, even the update went well and all important user information was kept.

Package Manager

While the package management system is now intelligent enough to



Plenty of apps and plenty of help for new users. And some cool games.

offer choices (e.g. "you need to install one of the following"), the actual dependency lists themselves are often found wanting. My pet hate with distros is *MySQL/PostgreSQL*. I only want one database thank you, and since we use *MySQL* for various stuff, I want it to be that one. Try not installing *PostgreSQL* or uninstalling it and you run into problems. Using the *Control Centre's* 'remove software' feature informs me that if I choose to do without *PostgreSQL*, I'll also have to give up *KDevelop*, *qt3-devel*, *kdebase-devel* and a host of other things. *Qt* may well have hooks for SQL, but it doesn't absolutely require them as far as I know.

Generally though, the new system works well. It is a debatable advantage to have two applications rather than one – while this is probably easier for new users, some complain about having to switch from one to the other all the time.

The software source manager is also separate and manages the list of install sources. One change here is that a list of cooker sources is not retrieved – if you want to add cooker updates, you'll have to find a site and add it as a normal ftp source (this is probably just as well, as adding loads of cooker updates is an almost certain way of increasing the flakiness of your system).

As with previous versions of Mandrake, it's the little things that make the difference. On noticing the Control panel had an option to configure scanners, I clicked on it to explore more. I hadn't actually installed any of the *SANE* software

What's included

Kernel 2.4.19
Glibc 2.2.5
GCC 3.2
XFree 4.2.1 / XFree 3.3.6
KDE 3.0.3
GNOME 2.0.1
BlackBox 0.62
OpenOffice.org 1.0.1,
KOffice 1.2
Mozilla 1.1
The GIMP 1.2.3

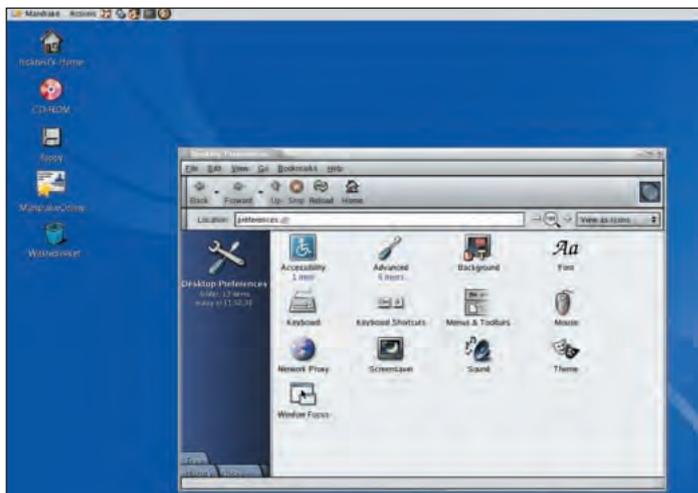
during the install, but the config tool realised this and automatically offered to fetch the packages and install them before proceeding. Little touches like this make Mandrake, more often than not, easier to use than contemporaries – a newbie to Linux probably wouldn't even know what *SANE* was, and thanks to this 'wizard', they don't have to.

What's in

Mandrake 9.0 includes both KDE and GNOME desktops, as well as other window managers should you choose to try them. Although efforts have been made to retain some degree of consistency, in terms of themes, icons, menus and so on, Mandrake haven't gone as far as the recent Red Hat and tried to merge the two together.

The KDE desktop seems to have had more attention paid to it. It is the default I suppose, but starting up into GNOME leaves you with a very bare desktop setup.

Of course, *OpenOffice.org* has been promoted to the chief office suite, and the various components have been



GNOME2 is included, but the default desktop seems quite spartan.

ReviewsMandrake9

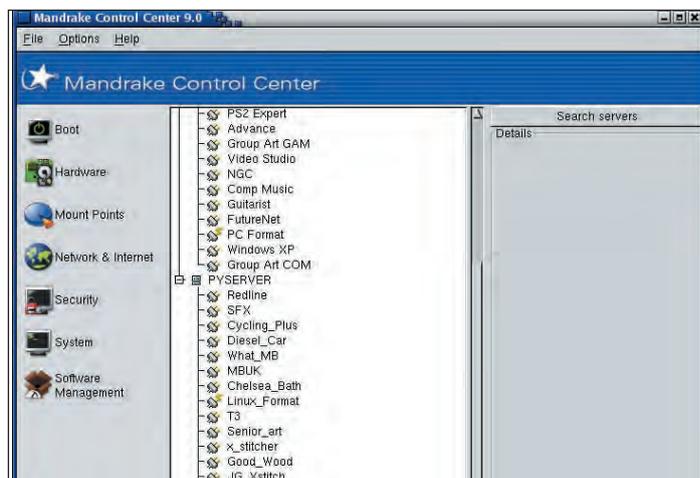
distributed to their natural resting places on the menus (although the *KOffice* tools remain as the defaults in the 'What to do' menu).

The main configuration tools in the *Mandrake Control Centre* have been significantly enhanced with the addition of more controls and features. One of the nice additions is the backup wizard. This isn't an enterprise scale solution, it's just a script, but it can handle backup of multiple directories and supports backup to network, hard disk, CD or tape drive, and can be scheduled. For the average user, an easy to use solution like this will be sufficient.

Problems

One area of Mandrake that seems to oscillate between working and not working is *Supermount* – the removable media mounter. It has a tendency to lose the ability to read media after a single access, requiring the disc to be ejected and reinserted. This is a shame, because it worked much better in 8.2, and we had hoped it was fixed.

There are plenty of other minor niggles. As many have mentioned on the forums, the *kdeartwork* package is not installed by default (and therefore the themes and screensavers that go with it), the install may recognise other versions of Windows, but doesn't notice other versions of Linux unless they are using the same bootloader, for some reason the resolution display panel forces you to logout before changing to a new depth – the list goes on, but most of these niggles are occasional headaches rather than seriously undermining the worth of the whole system.



Pay attention Red Hat. This is *Samba*, and this is how easy it should be to mount shared folders.

What You think

Vox populi - comments from the LXF forums

The *Linux Format* forums are full of people who are always trying out news stuff, so it was easy to locate a collection of users who had already tried out Mandrake 9. Here, briefly, are some of their (slightly edited) experiences:

Posted: Oct 31, 2002 - 02:06

Hi, My computer is a Gateway Solo 2550 laptop. Overall, I'm very happy with it. It looks better than earlier versions. Control Centre is better organised and smoother running, all of the window managers and desktops are well configured (though KDE3 has an annoying habit of emptying its menu from time to time). Setting up my network connections and network (over WinNT) printer were as straightforward as it gets. It also has no problem switching between the laptop keyboard and the USB MS Natural Keyboard I use to sit back from the screen.

My only real complaint is that I can't find a way to configure kdm, so if you want to use a window manager not on their list (I use fluxbox, which is great), you have to either log in from cli or configure a default automatic login.

Sylvan Kell

*I have installed Mandrake 9 on my desktop (800MHz Duron) and my laptop (Toshiba Satellite T100), and on the whole it seems good on both. Like Mandrake 8.2 before it, it required the **mem=nopentium** option to be added to Lilo in order to run properly with the Duron processor (though I have read*

somewhere that this may be more due to the KT133 chipset).

So far I'm very impressed by it. Installation is just a smooth and easy as you would expect from Mandrake.

First impression are that KDE3 is a lot faster than KDE2, especially when using Internet-related programs. One thing I don't like is on the desktop – rather than individual icon for CDRoms, etc., are now all grouped under a removable device icon, that then takes you to the individual device à la My Computer.

The Mandrake Control Centre and all their other configuration tools have also gone through a revamp which makes life far easier.

The only problem I had with installation is that shorewall, the easy to configure firewall package included with the distro, killed all traffic in both directions making the Internet temporarily inaccessible. I have had to find another firewall solution.

Reklan.

While there are some good improvements over 8.2 (for instance it starts up quicker) I wouldn't recommend anybody running 8.2 to bother with it, as there are too many nasty bugs in it.

There's no Kudzu or Kudzu replacement. So when you put new hardware on, there's nothing to detect it and guide you through installing the necessary software.

Bits are missing from KDE. For example, you only get about three

screensavers (not including kslideshow, which I like) rpmdrake is seriously screwed. If there isn't a disc in the drive when you launch it, it will spend several minutes trying to read it anyway, before finally realising something's wrong and prompting you to load a CD.

I don't like the way they have split rpmdrake into three separate tools for install, remove and upgrade (even the source manager is called separately instead of from within the main program now as well) – most inconvenient.

Tony Green

I installed Mandrake 9 on a Dell C800 Latitude laptop. It went in first time without a single problem. It auto-sensed the PCMCIA cards, the embedded NIC, screen, mouse etc.

The only thing I had to do was to install the nVIDIA drivers for the GeForce2Go video card. That worked first time as well. I use KDE3, Evolution, Galeon, and OpenOffice.org.

I use the laptop for development, so I'm running Apache, Resin (JSP engine), PHP, and MySQL. I've also got Samba running so that I can exchange files with my Windows box. I use JEdit to cut Java code.

The only minor niggle is the new software manager. The old version had "install" and "remove" as options in the same application. In Mandrake 9 they are both separate applications. Juggling software around is therefore more fiddly than before.

tenjin

Mandrake has always been more 'bleeding edge' than the rest of the major distros, and it seems unlikely things will change now. In defence of Mandrake, it has to be said that most

problems with Mandrake 8.0 (and there were a lot more) were sorted out within weeks of release. Configure your update source and you may find the problem is fixed before you come across it again – especially if you use the automatic update available to members of the Mandrake Club.

Conclusion

The Mandrake distribution hasn't changed so dramatically between 8.x and 9.0 as it did from the 7.x series moving up to 8.0. Some may say a full version increment is unwarranted. But there are loads of nice new touches in Mandrake. Sure, most of these are supplied by better Free Software running underneath, but the same is true of many distributions. There are no critical problems with this release (though the *Supermount* bug is a major pain) and we expect most of them will be sorted out by updates by

the time you read this (in fact, a problem with one wizard has already been fixed in the time between starting this review and finishing it).

It isn't rock solid, but it isn't tedious to use. It isn't transparently easy to use, but it doesn't restrict your choices. It isn't going to set the world on fire, but it is still arguably the best solution for desktop users. **LXF**

LINUX Format VERDICT

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

Maybe not progressing to the next level, but certainly holding it's own on the desktop.

LINUX Format **RATING**
8/10

REMOTE ADMINISTRATION SOFTWARE

NetOp Remote

If you need to administrate systems remotely, David Coulson may have the system for you.

Other remote administration tools including OpenSSH and VNC.

- **SUPPLIER** Richmond Systems Ltd.
- **PRICE** £139 ex. VAT for one guest and one host £650 ex. VAT for one guest and ten hosts.
- **WEB** www.netopuk.com
- **PHONE** 01428 641616

Remote access software is nothing new for Linux. We've been using everything from telnet and SSH through to X over TCP and VNC for years, but Windows has taken a while to catch up with us. Particularly with co-located servers, or for administrators or developers who choose to telecommute for some of their time, having a reliable and secure method to maintain systems from home, a laptop, or even a workstation sitting in the office saves on time and hassle, not to mention petrol to drive down to the data centre.

NetOp Remote is split into a number of different modules, which can be purchased either separately or

as a combination. The two main modules are Guest and Host, which are used to allow one system to access another. The Host module is executed on a server or other remote system and any other machine which runs the Guest module is able to connect to the Host. For each system which is to be controlled remotely, a Host licence must be purchased, and for each workstation which is to control a server running the Host module, the appropriate Guest licence must be purchased.

Guest module.

The most basic capability of *NetOp Remote* is to view and control a remote machine's desktop. For a Windows server, this is particularly useful, but how many people run X on a Linux server? Fortunately, there are plenty of clients available, so even if we have a Windows server, we can view the desktop in Linux and Windows. We are

also able to access a command prompt on the remote server through this software, which makes it much more useful for a Linux server. For any session, we can record all activity and then replay it at a later date, which might be useful if regular users are using the system, but for system administrators, it's likely not going to be used very much.

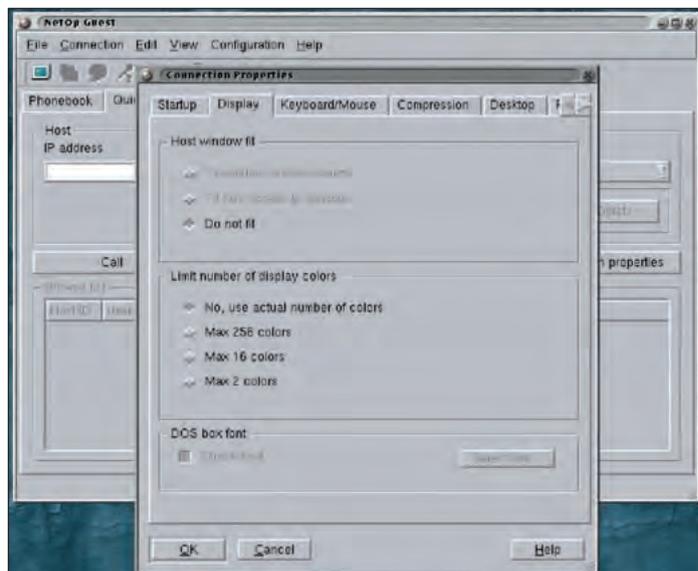
NetOp Remote's Guest module contains a rather nifty tree-based hierarchical view of the hosts available, so they can be organised into smaller groups depending upon a particular use of the system, access capabilities or even physical location of the box. This tree can be exported and distributed between clients, so if a new network is rolled out or if everything is

reconfigured, no one has to have the awful job of reconfiguring all the guest systems.

We can also copy files from the Host to the Guest system via *NetOp Remote*, which is particularly useful as it's quite uncommon for systems such as Samba or NFS to be implemented on network servers unless they are specific for this purpose.

Quite how this compares to *rsync* depends upon how comfortable the administrator is with the command line, but certainly it's not a major problem with any Linux system to copy or synchronise files between another system using SSH and *rsync*. *NetOp Remote* produces comprehensive transfer logs along with a tree-style view of the transfers, which is useful for keeping track of lots of files, particularly when dealing with synchronising configurations.

Arbitrary functions can be performed on the remote host via the Guest module, including issuing a **Ctrl-Alt-Del**, rebooting it or shutting it



NetOp Remote can be configured to allow a single guest to connect to a number of different hosts on a network.

down. However, if the server is locked up and requires a power cycle, then we're going to need more than *NetOp Remote* available to keep everything ticking over.

As good as *NetOp Remote* is, there are going to be numerous situations where you have no choice but to physically do something to a system, or need other software. Certainly one thing which springs to mind is issuing a **Sysrq** on a locked up system to debug the state of the kernel. As *NetOp Remote* doesn't have any physical access to the system, it can't replace remote management tools and consoles running on serial consoles, but it does go some way to filling the gap.

The Host module

Security is a very strong selling point of *NetOp Remote* and they've certainly gone to great lengths to ensure that their products will stand up to the current level of attacks and exploits. With an array of authentication methods, including their own *NetOp Security Server*, it's possible for users to be authorised for access through a number of different methods. It would have been nice to see LDAP or NIS included, although with the appropriate modules, it is possible to allow users to authenticate via these methods using PAM.

From a network standpoint, there is a selection of security methods implemented, including filtering based upon MAC and IP address, but one may still choose to consider running the connection over a secure VPN system, just to be on the safe side. If the *NetOp Security Server* is used, the capabilities of individual Guest

users can be modified depending upon their rôle in the organisation, so you don't have to worry about giving someone unrestricted access to a server just to perform a simple function remotely.

On the administration front, everything is logged by *NetOp Remote* and is available in the operating system's log system or via SNMP by a remote client. Having it all available via SNMP makes it easy for a large number of hosts to have their SNMP data aggregated for analysis or otherwise combined to ensure that no unwanted access is missed by administrators.

Interestingly, *NetOp Remote* will function over a variety of networking protocols, including TCP/IP, UDP/IP and IPX, not to mention it can handle a wide selection of physical networking options, making it ideal for nearly any network environment. It will even work happily over a serial connection, a PSTN or ISDN dial-up connection or even over IrDA without a network protocol being available. This boosts security significantly, as if we're not accessing the remote system over a public network such as the Internet, then the options available for an attacker are fairly limited, although quite what they were thinking when someone decided to check that it worked with IrDA is anyone's guess.

An interesting addition is the inclusion of asset management software, which seems a rather strange inclusion in a package of this type. Whether this was originally supposed to be a separate software package which was to be distributed separately and was just thrown into *NetOp Remote* is a mystery, but there are many other



NetOp Remote can run on a wide variety of platforms, allowing for cross-platform connectivity.

reputable asset management packages available and one would assume that this would be a separate consideration from remote management.

Déjà vu?

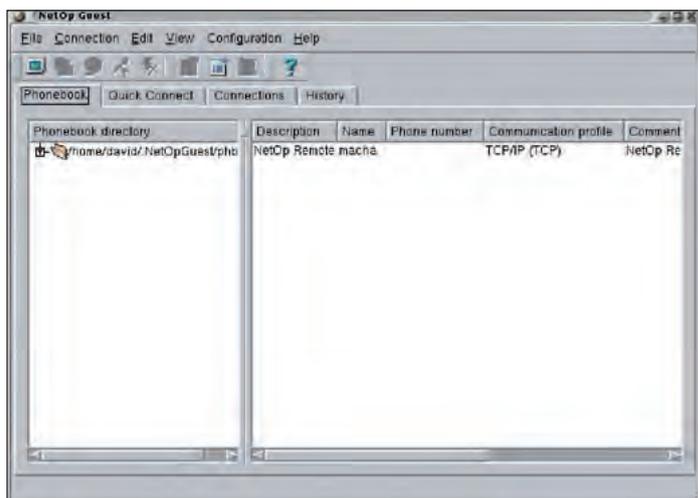
Maybe I've missed the whole point to this package, but there really isn't anything new here and experienced Linux administrators will likely not have any great need for *NetOp Remote*. Almost everything which it does can be performed with Open Source and freely distributed software, making it all somewhat redundant in the world of Linux. If Linux users want remote X access, there is VNC, which is available natively for nearly every OS known to man, or for everything else via a Java client. As much as people complain about Java, I'd sooner take a Java client I can run on anything over an ActiveX client which requires *Internet Explorer*. VNC, originally developed by the AT&T labs in Cambridge, is a truly fantastic piece of software – and for low bandwidth environments, there is the *TightVNC* patch, which has particularly effective compression capabilities, making it useful for almost everyone. For console access, there are more protocols than you could shake a stick at, and *SSH* pretty much has a monopoly over remote system access, particularly over a public network such as the Internet. With such an unrestricted license and regular security updates, *OpenSSH* is a near perfect choice for almost all remote access purposes, not to mention that it has a reasonable track record with

security, along with a massive user base to report bugs.

Conclusion

Maybe people feel more confident with a commercial product, but so many Linux distros come with packages which meet the majority of the expectations of this. For high-end networks, KVMs are now available which provide remote access without anything being installed on the server, which is clearly much more useful, as if a system locks up or network access is unavailable, then even the best remote access software is going to be useless.

It's hard to see who would need *NetOp Remote*, unless they had a Windows network and wanted some of the more fancy features it offers over, say, VNC. Short of needing the guest side installed to administrate Windows system, there is little here for admins of Linux-only server networks. Some of the more fancy connectivity capabilities might be useful for some, but with the flexibility of a Linux system, it's not too difficult to implement with Open Source solutions. **LF**



The guest side of NetOp Remote allows us to connect to remote systems.

LINUX Format VERDICT

Ease of use	8/10
Features	9/10
Performance	8/10
Value for money	4/10

A great remote administration tool, but more suitable for Windows users.

LINUX Format RATING

7/10

VIRTUAL MACHINE

VirtualPC 5

Running Linux on Windows sounds like a retrograde step. But, **Andy Chappelle** discovers, it has its uses and could help bring Linux, OpenOffice *et al.* to the World's office-bound masses.

Both VMWare and Netraverse have competent – and native – virtualisation software.

- **DEVELOPERS** Connectix
- **PRICE** £193.87
- **WEB** www.connectix.com

Connectix is no stranger to x86 emulation; their flagship product, *VirtualPC*, has been regarded as the best way to put Windows and Windows apps onto a Mac since its premiere release. However, on the release of version 4, the company took the step of offering x86 emulation on Windows, allowing users to install and run a variety of OSs on a single Wintel machine. *VirtualPC* creates, as expected, a virtual machine within your OS, but unlike products such as *VMware*, it emulates almost every element of the virtual machine from the sound and graphics subsystems to the network interface.

Who wants it?

VirtualPC will never be an impulse buy, but there are specific uses where the speed and flexibility of *VirtualPC* will save time and money over dual-booting or using *Win4Lin* or *Wine*. Connectix have wisely noted the growth of multi-OS IT services and cross-platform app development, and are pitching *VirtualPC* as a way of supporting Windows and a variety of Linux flavours without having three or four machines on every coder's/helpdesk operator's desk. The installation is a standard *Installshield* affair and involves the inevitable Windows reboot. It did fail to spot a pre-existing Linux (Mandrake) image in my 'Connectix' directory (this is probably forgivable, as I'd removed *VPC4* to make way for a clean install of its replacement), but had no problem loading it after the install. Not all was well though. While the distro image survived, its network config didn't, and reinstating it wasn't as straightforward as you might have thought. That said, it wasn't long before I had both host and guest OS connected to the 'Net which made checking a web project worked

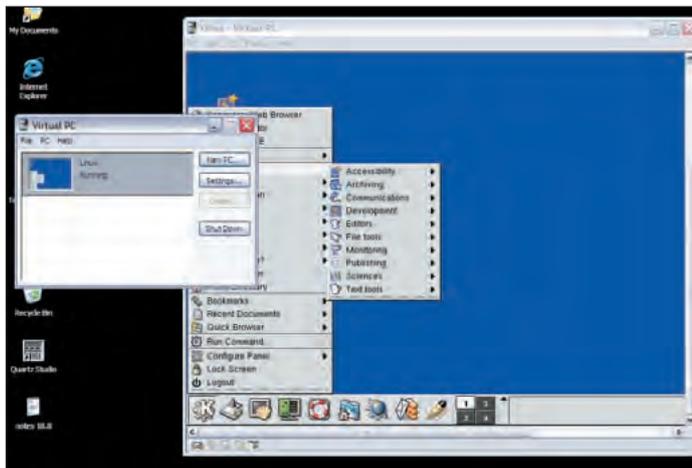
as expected with *Mozilla*, *Konqueror*, *Opera* and *Internet Explorer* at the same time a breeze. As I'd just added extra memory to the host machine, I was also able to also virtually upgrade the Mandrake installation, simply by sliding the memory configuration bar to the right, and then restarting the machine. Essentially you can run as many OSs as your hardware will handle. *VPC* demands 128MB RAM for the host (in the case of WinXP) and 64MB (plus 2GB of disk space) per version of Linux. However, you can run a basic Linux system in as little as 8MB/100MB meaning you could theoretically run a fairly large number of virtual machines at once. But why would you want to? Well, this is where *VPC*'s real strength lies. It's a convoluted operation involving creating a loopback network device on Windows XP (this feature won't work on previous MS OSs) and then manually assigning IP addresses to your guest systems, but within a short time, you can install and configure your own virtual network, with guests able to access each other and the host. You can also remotely control a guest OS using any standard VNC client. This is fantastic for network testing, and with the 'Virtual Switch' networking you can opt to let each machine access local, host and external connections – or all three – allowing you to cut one machine from the net with relative ease. One of the criticisms

of previous versions from Linux users was the lack of machine-to-machine communication, but with virtual network switching on, that's no longer an issue.

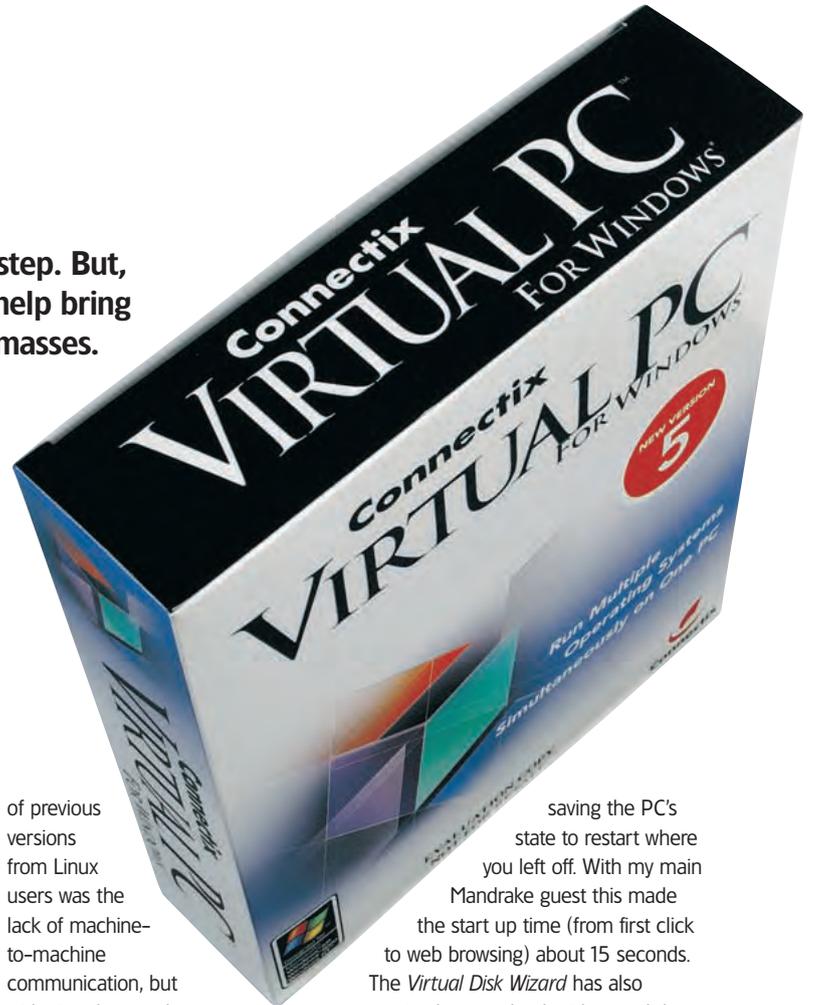
Faster Pussy Cat

Upon upgrading the emulated OS feels faster now. With the guest in full screen mode (given a 1GHz+ PC) Mandrake is a little quicker than a 500MHz PII in my unscientific tests (rendering image effects in *The GIMP*) and makes for a perfectly useable system. Also on the subject of speed, start up and shutdown times for a guest OS are comparable to running native, but when you shut down an OS, you have the option of

saving the PC's state to restart where you left off. With my main Mandrake guest this made the start up time (from first click to web browsing) about 15 seconds. The *Virtual Disk Wizard* has also received an overhaul, with new disk types beyond the expandable and undoable (which can be set back to its initial state with one click) drives. *VPC* now has a difference drive, which is linked to a second drive image but only the data that has been changed is saved to the disk, leaving the original drive intact, and a pair of partition drives. Overall this is a great product. It should make helpdesk support across platforms easier, thus helping to spread the joy of Linux. It also makes a great testbed for cross-platform application development and network testing. Now all we need to make things even better (and this is not out of the realms of possibility) is a Linux version! **LXF**



Improve your Windows desktop, by running Linux on it.



LINUX Format VERDICT

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

Robust virtualisation software with excellent networking capabilities. Worth a purchase if you or your employer are going multi-platform at any time in the future.

LINUX Format **RATING**
8/10

SCANNING SOFTWARE

VueScan 7.5.60

Hoyt Duff's journey of discovery results in an enchantment with an application that really does make scanner access easy.

Extremely capable proprietary, cross-platform rival to SANE, with good support .

- **DEVELOPER** VueScan
- **WEB** www.hamrick.com
- **PRICE** US\$40

Sometimes, the most fun part of Linux is in the serendipitous discovery of new things.

That's one aspect of Linux that holds some attraction for many of its users, us included. Since access is granted to the inner workings of the operating system and much of the software, it can be interesting and challenging to make a piece of hardware work when it doesn't – at first blush – appear to work at all.

Such was the case when I was involved in a recent writing project. Needing to write about *SANE* and scanner support, I attempted to use a Canon N650U USB scanner that had been connected to a Windows computer used by the lady of the house.

Free scanner support in Linux is provided by *SANE* (*Scanner Access Now Easy*), an API (Application Programming Interface) for scanner hardware. *SANE's* goal is to provide free drivers for scanning hardware that can be used by any appropriately-

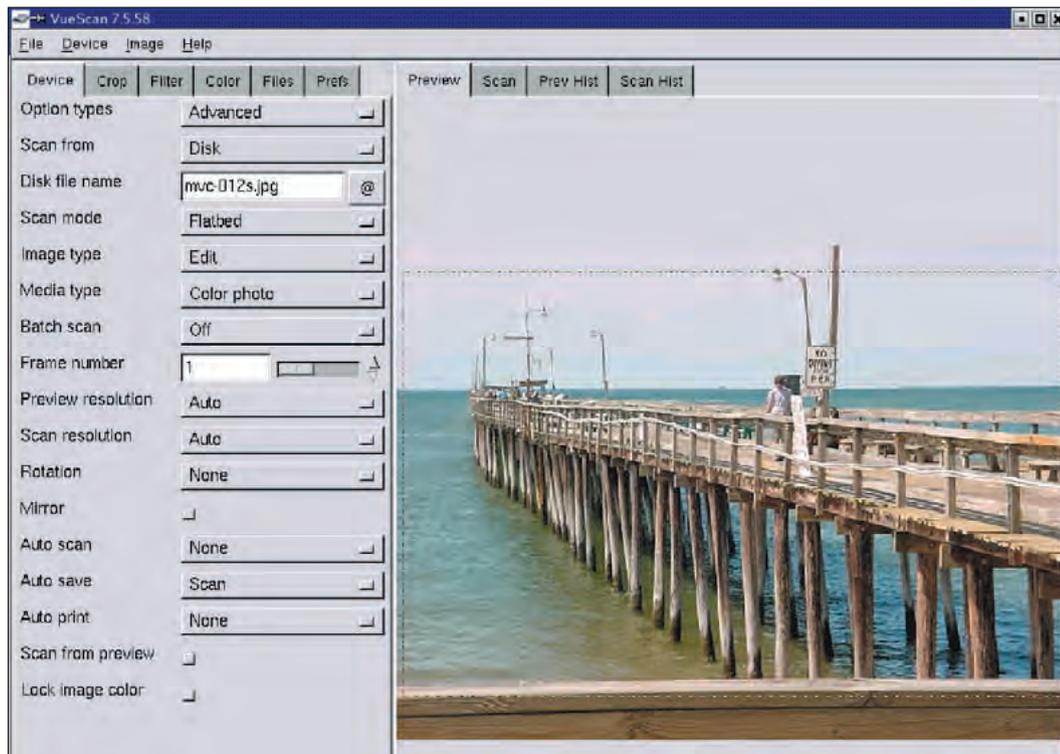


Fig 2 **Advanced options are very powerful. Basic and intermediate settings are also available.**

written application. This is a little different from *TWAIN* (it's not an acronym, but actually an archaic word meaning two items of the same kind; some people jokingly – and incorrectly – refer to it *Technology Without An Interesting Name*), the

approach used in the Mac and Microsoft Windows. While the goals of *TWAIN* appear to be the same as *SANE's*, the *TWAIN* design integrates the driver and application, forcing the driver to use the graphical API instead of being independent. This is why Windows drivers are of no use in Linux as *TWAIN* drivers need the Windows or Mac API. Curiously, *TWAIN* could access a *SANE* driver. The reason that *TWAIN* is not implemented in Linux is the dependence on a particular graphical API (Linux has several, Mac and Windows only one) and the difficulty in having *TWAIN* work over a network in a manner easily compatible with the Unix way of doing things.

SANE is organised into "back-ends" and "front-ends". The back-ends are the device drivers that access the hardware. The front-ends are the applications that the user interacts with. Because of this dichotomy, any application can easily incorporate the use of *SANE* drivers. Most notable

among the front-ends is *The GIMP* plug-in that permits scanning directly from *The GIMP*. There are similar plug-ins for *StarOffice/OpenOffice.org* and other applications as well. The KDE scanning application *Kooka* incorporates *SANE* support and uses the *gocr* application for OCR (Optical Character Recognition).

The *SANE* website at www.mostang.com/sane provides *SANE* documentation, binaries and source code for download, and most importantly, a searchable database of supported scanners. As with any hardware that you intend to use with Linux, it is always preferable to consult such a list before you purchase any hardware. In my case, I had owned the scanner for some time and did not want to purchase another.

To my sorrow, I quickly discovered that the Canon N650U USB scanner was not supported by the version of *SANE* included with Red Hat 8.0 or Mandrake 9.0 (version .84 and .87,

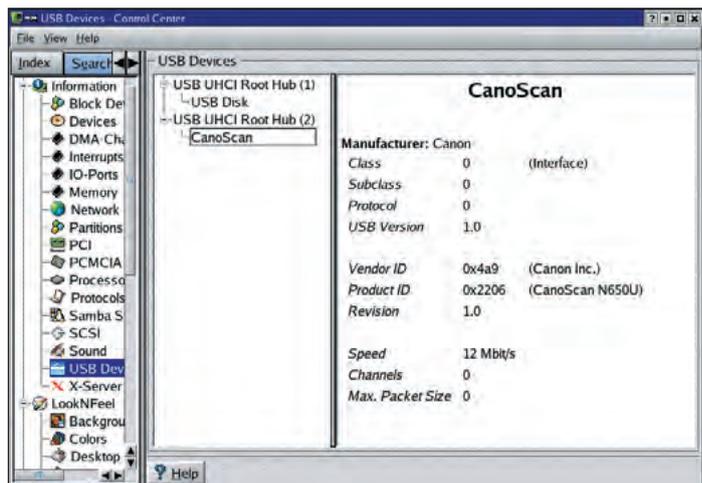


Fig 1 **Nicely displayed USB information from KDE's KControl application.**

respectively). As of this writing, it appears that the most current version available (version 1.09) does provide limited support for scanning in 8-bit colour. Version 1.09 is only available as source code, which is freely available from <ftp://ftp.de.mostang.com/pub/sane/>, the German FTP mirror.

Not having official support didn't stop me from making an attempt to use the scanner, however, figuring that I would learn a little bit about how USB devices were detected and their kernel modules loaded and configured.

Knowing that the scanner worked in Windows, I plugged it into the USB port of our Mandrake computer. Modern Linux kernels use the file `/etc/hotplug/usb.distmap` to automatically detect and load the appropriate USB driver for a device listed in the file. It's part of the HotPlug support included in the kernel since January 2001. The Canon N650U is not listed, so the appropriate kernel device driver won't be automatically loaded. The next step was the kernel documentation at `/usr/src/linux/Documentation/usb/scanner.txt`, to see what was written about the USB scanner kernel module. There, we learned that we should load the scanner module in the following manner:

```
scanner vendor=0x####
```

```
product=0x****
```

This meant that we need to know the Vendor ID and the Product ID, which is not something that is typically included in the (now long lost) scanner documentation. Linux provides several avenues to obtain the information, and while it can be obtained from the output of `cat /proc/bus/usb/devices` as shown here (abbreviated):

```
cat /proc/bus/usb/devices
```

```
T: Bus=02 Lev=01 Prnt=01 Port=01
Cnt=01 Dev#= 2 Spd=12 MxCh= 0
D: Ver= 1.00 Cls=00(>ifc ) Sub=00
Prot=00 MxPS= 8 #Cfgs= 1
```

```
P: Vendor=04a9 ProdID=2206
Rev= 1.00
```

```
S: Manufacturer=Canon
```

```
S: Product=CanoScan
```

```
C:* #Ifs= 1 Cfg#= 1 Atr=80
MxPwr=500mA
```

```
I: If#= 0 Alt= 0 #EPs= 3
Cls=ff(vend.) Sub=00 Prot=ff
Driver=(none)
```

```
E: Ad=81(l) Atr=03(Int.) MxPS= 1
Ivl=16ms
```

```
E: Ad=82(l) Atr=02(Bulk)
MxPS= 64 Ivl=0ms
```

```
E: Ad=03(0) Atr=02(Bulk)
MxPS= 64 Ivl=0ms
```

You can see that there's a lot of confusing information presented by the command. The information pertinent to the Canon N650U is highlighted above.



Fig 3 Yes, it runs on OS-X, too – for all those of you who haven't got round to trying Debian PPC on your Macs.

Note that `Driver=(none)` tells us that no driver is loaded. The line `P: Vendor=04a9 ProdID=2206 Rev= 1.00`

We could have more easily got that same info from the `lsusb` command:

```
lsusb
Bus 002 Device 001: ID 0000:0000
Bus 002 Device 002: ID 04a9:2206
Canon Inc. CanoScan N650U
```

As well, the *KDE Control Panel* can be used to obtain the same information as shown in the **figure 1**.

We can manually load the kernel modules with the following command: `modprobe scanner vendor=0x04a9`

```
product=0x2206
```

To make this happen automatically, we also learn from `/usr/src/linux/Documentation/usb/scanner.txt` that if we insert the line

```
options scanner vendor=0x04a9
product=0x2206
```

into the `/etc/modules.conf` file, the kernel module will load correctly every time we restart the computer. How convenient for everyone.

To verify that the module is indeed loaded and working with our scanner, we again run the command:

```
cat /proc/bus/usb/devices
```

Now we observe the `Driver=` line for our scanner and if we have been



Brightness

Controls for photographers



Fig 4 Note the lack of detail in the shadows under the building.

Figure 4 shows a late morning shot of a beach cottage in Nags Head, North Carolina, USA. Note the lack of detail in the shadows under the building.

A change to the Brightness setting

and the addition of the fading and sharpening filters from the Filter tab, and the image looks much better (**fig 5**).

The features offered by *VueScan* appeal more to the photographer. For



Fig 5 After the changes...

example, the manipulation of brightness is available to allow the photographer to control the level of detail in light and dark areas of the picture that are typically munged by

the film lab during processing. The options to select the film type when scanning from colour or black and white negatives is also targeted toward those users.

ReviewsVueScan

◀ successful, it will look like:

Driver=usbscanner

However, *SANE* still will not work with the Canon N650U scanner because the installed version does not support it. If we downloaded, compiled and installed the current version of *SANE*, we would still only have support for 8-bit colour scanning. This is encouraging for the *SANE* project, but not acceptable for our use as the Canon N650U scanner is capable of scanning a 1200x2400dpi, 48-bit image.

Continuing our efforts, an Internet search using www.google.com/linux to look for any other available support for our scanner led us to <http://homepage.ntlworld.com/millwardjp/> where we found a stand-alone Canon N650U scanning application.

There, Jason has written an application that will allow him to use his Canon N650U with Linux. To use it, you must download the source code from <http://homepage.ntlworld.com/millwardjp/n650u-v2.0.tgz>.

Extract the files from the archive and follow the included instructions. Unlike much compiling you may have done in the past, you will be compiling the source code file in the following manner, as there aren't any makefiles:

cc N650U.c -o N650U -Iusb

This uses the *GCC* compiler to create a binary named *N650U* that can be run with arguments to set the dpi and it will create an A4-sized scan as the file *image.ppm*. That image can be manipulated with the *NetPBM* tools and converted into a more suitable format if needed. While Jason is to be commended for his considerable efforts, that is a bit of work to get a scanned image. (He includes a scan of part of a *Linux Format* page as an example of his efforts!) Hopefully, continued development of the *SANE* driver will allow full use of the scanner.

Still not to be deterred, we stumbled across a commercial scanning application that

- 1) allows us access to the full features of the scanner (including advanced features to high-end scanners);
- 2) is reasonably priced;
- 3) is well-supported; and
- 4) comes with a very liberal licensing policy.

The documentation is also well-written and the website informative and useful.

And so begins our review proper...

VueScan

You can download and demo the most current version of *VueScan*, but the demo cannot save images in raw format and it places a watermark on the image. When you purchase a serial number, those restrictions are removed. As well, you may use any version of *VueScan* on any four computers that you personally use. You receive unlimited free upgrades. Multi-user licenses are available: a ten-user license for US\$250 and a site-license for US\$2,000.

After downloading the .tgz file, extract the files with:

```
tar zxvf vuescan75.tgz
```

which will produce the following files:

vuescan.bmp	# The Vuescan splashscreen
vuescan.dat	# The VueScan film data file
vuescan	# The VueScan binary
vuescan.htm	# The VueScan release notes
/html	# The User's Guide

It is not necessary to relocate the files in order to run the application, but you'll want to create a desktop link or add *VueScan* to your menu. That's not difficult to do, but the inclusion of an .xpm icon for the link would be a nice touch. Launching the application is trouble-free, unless you are using Red Hat 8.0, where the language environment needs to be set first or the fonts are corrupted; see the *Linux Release Notes* for the details.

When you run *VueScan*, it will generate a *~/.vuescan.ini* file that contains the settings you select for the application. When you purchase a serial number, that will be automatically placed in *~/.vuescanrc*.

While we don't have a Mac to test on, an associate of ours has satisfactorily used the Mac version for a few years. The Windows version required that we install the Canon drivers first, but worked in an identical manner to the Linux version.

When at work, *VueScan* performs two different actions: scanning an image and processing the scanned image. After scanning an image, *VueScan* keeps the image in memory as a "raw image," which is the raw CCD data from the scanner itself without any colour correction or scaling. This is a useful and important part of how *VueScan* works. Any manipulation of the raw data is not made to the raw data itself,

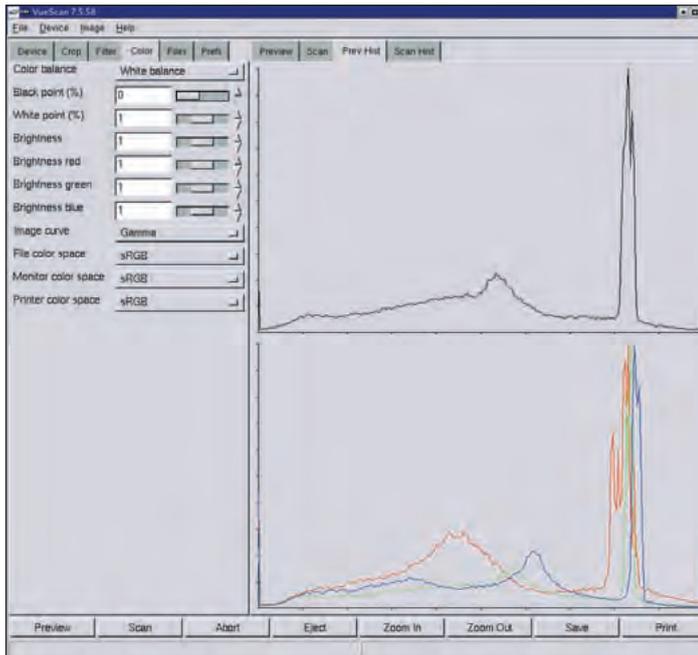


Fig 6 The colour correction tab and the associated histogram.

but to data that is held in a separate buffer. This means that you only need to scan an image one time. You may save the raw data and reload it later, just as if you had scanned it in anew.

Manipulation

The processing done is not the kind most commonly associated with an image manipulation program like *The GIMP*. Focusing on providing excellent colour from scanned colour negatives, *VueScan* provides the ability to use the infrared data (if available on your scanner) to remove dust spots, then correct for colour shifts and dye fading, perform corrections for film media, sharpen the image and finally correct the colour. When you are scanning from film, the resulting image depends on the kind of film used, *VueScan* provides choices for several hundred different types of film and allows auto-correction for black and white negatives, colour-process black and white film and colour film.

The interface as shown in figure 2 presents the Advanced Options. Also available are Basic and Intermediate settings. Since it's provided with excellent default values, it is suggested that you begin with the Basic option and experiment before advancing to more involved options. The final image can be saved in TIFF or JPEG format; the raw data remains untouched. Use *The GIMP* for any processing or manipulation other than the initial colour correction.

Figure 6 shows the colour correction tab along with the associated histogram. The white point setting is perhaps the most useful to bring out the details in the image. You may also observe the changes in the Preview as you change the values.

Support

Support is available in the newsgroups and the *VueScan* provides a link to the search engine for Google Groups. A site that includes a review of *VueScan* also has some very good basic scanning information; it's at www.scantips.com.

If your scanner is not yet supported by any *SANE* back-ends, you have a high-end scanner whose features are not supported by any *SANE* front-ends, or you have a need for an application that provides specialised support for scanning film, *VueScan* is certainly worth a look. [LXF](http://www.linuxformat.co.uk)

LINUX Format VERDICT

Installation	9/10
Documentation	10/10
Features	10/10
Ease of use	9/10

If you don't mind paying for a full-featured scanning application, need cross-platform support or advanced features, give this one a try.

LINUX Format **RATING**
9/10

Practical Python

Nick Veitch gets some Python practice

- **PUBLISHER** Apress
- **AUTHOR** Magnus Lie Hetland
- **ISBN** 1-59059-006-6
- **PRICE** £30.00

Python is certainly growing in popularity, at least judging by the number of books now published on the subject! This particular volume gives the impression that it's more about doing 'serious' stuff in python rather than being a beginners guide, but in fact the basics of the python language are covered in great depth. This is actually explained very well, with plenty of relevant examples. The examples themselves are well thought out to include a range of uses and situations, not just the common or simplest ones – which is more than most books of this type

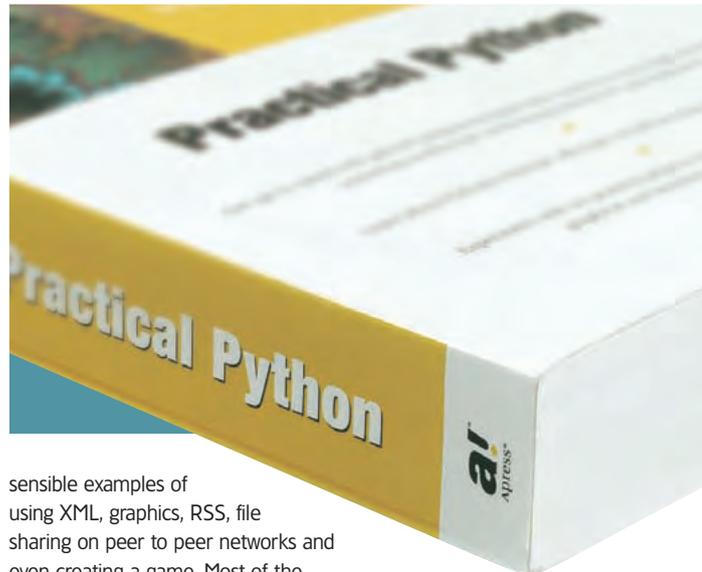
manage in my experience.

The section on the concept of Abstraction is particularly well done, and there are plenty of pointers for people of different abilities scattered throughout the text.

I'm not a great fan of the layout of the book. The pages seem quite spaced out, and a lot of the example code fragments wrap over pages which can be a bit annoying when you are trying to follow what's going on.

A good plus point is the discussion of non-core python functionality. While the standard python modules are really good, for a lot of projects it is really useful to use third-party modules. This mostly comes to the fore in the chapter on building python GUIs, but is also addressed in some of the example projects.

The projects themselves cover



sensible examples of using XML, graphics, RSS, file sharing on peer to peer networks and even creating a game. Most of the projects are based around using python as a web/server technology, which probably is appropriate. The essentials of XML could probably be covered in more depth, but this is an excellent introduction to real world python use.

Linux Format VERDICT

A good practical introduction covering some advanced topics

LinuxFormat **RATING**

8/10

Writing Perl Modules for CPAN

David Coulson jumps on the Perl bandwagon and takes a look at a CPAN guide for developers.

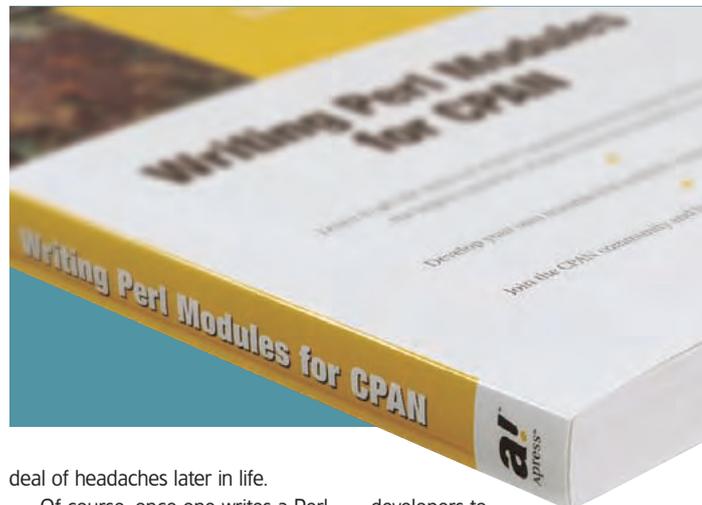
- **PUBLISHER** Apress
- **AUTHOR** Sam Tregar
- **PRICE** US\$34.95/ £23.30
- **ISBN** 1-59059-018-X

There is a huge range of Perl modules available, nearly all of them distributed via the popular Comprehensive Perl Archive Network, otherwise known as CPAN. In true Open Source style, one can contribute modules back to CPAN, as well as download existing ones for use by Perl scripts.

This book covers every aspect of Perl development when it comes to creating modules, whether they are distributed on CPAN or not. Starting out with some Perl basics and slowly moving towards handling Perl modules, anyone who is even slightly familiar with Perl will easily expand on the examples given in the book and can quickly produce modules for Perl.

Even for a Perl novice, the book has many easy to understand examples as well as some basics on Perl, so anyone with even the smallest amount of Perl knowledge can produce some usable code.

CPAN is quite a complex beast, yet it is covered in significant depth without becoming too confusing. For those just wanting to use existing modules from CPAN, installing and configuring the 'cpan' utility, or downloading them via HTTP or FTP, is indispensable. CPAN, being CPAN, has a number of standards which are all covered, including building your own Makefile.PL script, as well as specifying the correct information within the module so that it all installs happily. Not only are specific requirements looked at, but other suggestions which are simply good practice are included, and anyone starting out with Perl modules will find that getting into the right habits early on will save a great



deal of headaches later in life.

Of course, once one writes a Perl module, it needs to be kept up to date, so handling patches submitted by third parties is covered, as is dealing with versioning using cvs. These are great for module developers, but equally apply to anyone writing code in any language. A significant section of the book is also set aside for those developing Perl modules in C, and offers a great number of examples for those unfamiliar with certain aspects of the C language.

In all, this is a wonderful guide for anyone writing modules for Perl, whether using CPAN or not. While a substantial part of the book is reserved for CPAN, the fact that it encourages

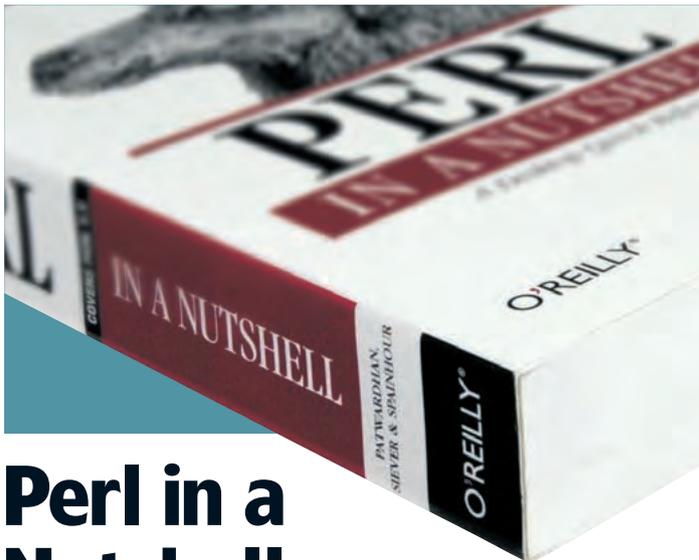
developers to build their Perl modules around popular standards and shows them how to correctly maintain that module, it is ideal for those who are starting out with development and are unsure quite what is the best way of approaching it.

Linux Format VERDICT

Everything one needs to know when it comes to developing and distributing your own Perl modules, but it doesn't just stop at the code.

LinuxFormat **RATING**

10/10



Perl in a Nutshell (2nd Edition)

David Coullson takes a look at a reference guide for Perl hackers.

- **PUBLISHER** O'Reilly
- **AUTHOR** Patwardhan, Siever & Spainhour
- **ISBN** 0-596-00241-6
- **PRICE** £28.50

O'Reilly have always been a particularly popular publisher when it comes to publications covering Perl, fondly known as 'the Camel books' by those in the know, as the cover of all their Perl books have a camel adorning the front cover. The second edition of *Perl in a Nutshell* includes numerous updates, as well as keeping up to date with the latest Perl releases by including version 5.8.

If you're just starting out with Perl, or are thinking about learning it, then you should take a look at the *Programming Perl* book, from the same publisher, as you will find yourself thrown in at the deep end, head first, if you are thinking of starting out with this type of reference book. However, for anyone familiar with Perl, or even experienced Perl developers, it's ideal when things get a little confusing, or we can't remember arguments for a particular function.

While it starts out with some basics of Perl, such as general syntax and dealing with variables, it is very much a refresher, rather than

something one could easily learn from, as the examples are very disjointed, and rather than having a complete Perl source to play with, there is just a little snippet of code. It even covers installing Perl on a wide range of platforms, including Linux and Win32, and looks at installing Perl modules via CPAN, as many modules which are not included as part of the standard Perl distribution are covered.

Two large sections of the book are devoted to Perl on Win32 and Perl/Tk, which is probably one of those things which people will either use all the time, or not at all. Why Perl/Tk was included over, say *GTK+*, is anyone's guess, but it's always handy to have a reference for when a quick and simple GUI is desired.

Everything in this guide is available online, or via *perldoc*, but anyone who is seriously using Perl, or simply wants a quick way find what function they need to use, will likely spend most of their life with this book sitting on their desk next to them. [LXF](#)

Linux Format VERDICT

Almost everything a Perl developer needs, but not the best choice if you want to learn Perl.

LinuxFormat RATING

/////// 8/10

www.linuxformat.co.uk

BX Pro

for Linux

Builder Xcessory Pro (BX Pro) is the industry's most advanced Graphical User Interface development tool.

BX Pro will help you quickly deliver intuitive, effective and robust Motif user interfaces.

BX Pro for linux includes Viewkit 2.1 an integrated C++ object re-use framework with a large number of C++ components to jump start your GUI development.

For more information contact Scientific Computers

<http://www.scl.com/linuxformatad/>
info@scl.com

tel: 01293-403636

Roundup

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



OUR SELECTION AT A GLANCE

- Anjuta
- Code Forge
- KDE Studio Gold
- KDevelop
- Kylix Open Edition
- Code Crusader
- GUIDE

C++ IDEs

Maurice Kelly packs away Vi, and tries a more integrated approach to code development with the latest batch of Linux C++ IDEs.

For many people life is always about making things that little bit easier for yourself. Imagine the early programmers having to input the basic machine code of their computers by hand – no mean feat for anything more than basic programs. It is little wonder that they soon created tools to make the programming process easier – assemblers made it easier to write programs and these were followed by compilers. It didn't stop there though – computer programs are becoming more and more complicated as time goes on, and as a result the desire to make programming easier still yielded Integrated Development Environments (or IDEs.)

For the uninitiated, an IDE is a consolidation of a number of utilities used in developing applications. The standard programming setup could involve the use of an editor, a file manager, a compiler, and (optionally) a debugger. Some IDEs can consist of all of these utilities, whilst some simply bring separate utilities together in one place, tying them together for the

convenience of the developer.

In this roundup we will be looking specifically at C++ IDEs. While C is the accepted base language of the Linux world, C++ is becoming increasingly popular, partly due to it being an object oriented programming language. Its popularity is also increasing as a result of the fact that the most popular desktop environment

for Linux is written in C++. KDE is the favoured platform for writing GUI applications and so demand for C++ environments is also increasing.

Due to the diverse nature and demands of many Linux users it can be difficult rounding up any type of application. However rounding up an application class such as IDEs is always going to be that little bit harder. Every developer likes to do things that little bit differently, and a development environment is often a finely tuned setup. Some IDEs are fixed in their operation which affords for a tightly integrated and reliable system. Others allow you to pull together a variety of components which allows for maximum flexibility and choice, but can result in less reliability. Configurability therefore is a primary concern, but stability and ease of use are also high priorities.

Before anybody says anything, GNU Emacs/XEmacs aren't being considered here as development environments. Everyone knows that they are really operating systems in their own right!

'Imagine inputting machine code by hand, it's no wonder that tools were created to make programming easier'

Anjuta

Powerful GNOME IDE, with a rough exterior.

- **VERSION** 1.0-beta1
- **WEB** <http://anjuta.org>
- **PRICE** Free

Anjuta is an up-and-coming C/C++ developing environment, based on the GNOME environment. As with any developing GNOME application, it's always advisable to have the latest libraries installed to ensure compilation/installation dependencies are satisfied. For those looking for pre-built binaries on the 'Net, the version reviewed was described as 1.0-beta1, but it was in fact version 0.99.9, so you may want to search for that version instead.

First impressions of *Anjuta* are not good – it initially appears to be very rough around the edges, and feels like a project very much under development. For an application that is supposed in it's "beta1" phase it lacks quite a bit of polish. The bare initial interface does not help, as it looks more like a basic GNOME application generated by *KDevelop* than a fully fledged developer studio!

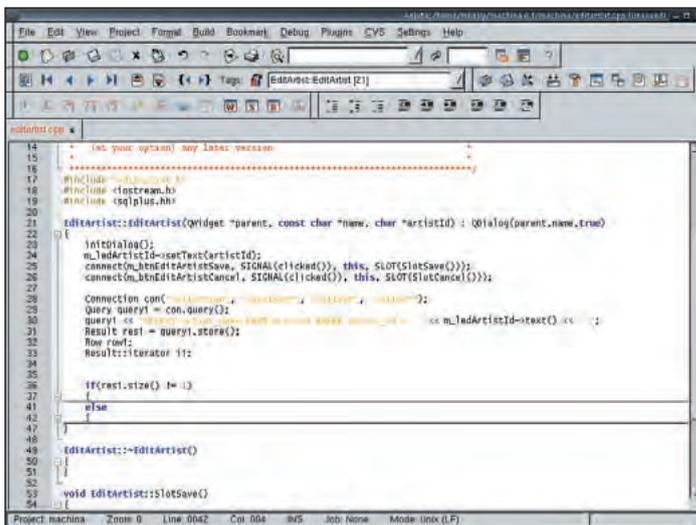
Still, if there's one thing that using Linux teaches a reviewer it is that you should never judge an application by how polished-looking it is and a quick delve reveals that *Anjuta* has a lot of features hidden under its skin.

Selecting to create a new project spawns a wizard that allows entry of some project information and allows you to select from a number of project

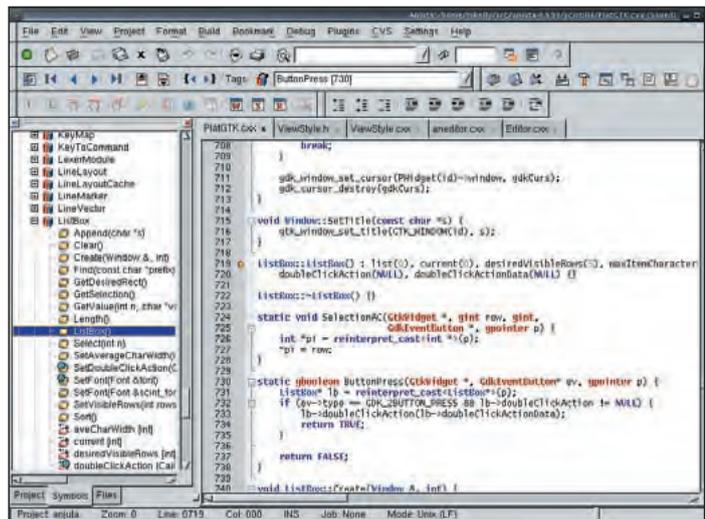
types. There is a large selection of project types, but they all very much follow a GNOME/GTK theme. Worth mentioning though is the ability to create a *wxWindows* project, as well as an *Xlib*-based application. If you want KDE-based apps you'll have to keep wanting or hack together your own from the basic terminal-based project. Although this is really a piece of GNOME software and it would have been nice to see the ability to create KDE apps included in some form.

Once a project has been created (or loaded) then we really see what *Anjuta* looks like, as the entire application comes to life. Like most development environments it takes the three pane approach – status windows across the full width of the bottom, editor pane across two-thirds of the top, and the remainder is occupied by tree views of the projects symbols and files. Just about everything in the application is detachable so you can plonk windows all over your desktop if you desire. A decent range of toolbars is available, and includes my own personal favourite – a separate debug toolbar!

The *Anjuta* editor is built-in, and there is no option to use your own editor of choice (unless you edit the files in your own editor separately.)



The *Anjuta* built-in editor with syntax highlighting. The code folding option can be seen in use as the black horizontal lines beside the gutter control.



Browsing some of the classes used within the *Anjuta* source code itself.

That said, it is a competent little editor and is packed with plenty of developer-friendly options. Particularly nice was it's code "folding" option – the gutter shows a small vertical line (capped with a box containing a minus symbol) that indicates the extent of code between a set of curly braces. Clicking the minus symbol "folds" (or collapses) the code inside the braces down so that it is hidden from view, allowing for more code to be seen at once.

A full range of debugging options was available, although there didn't seem to any way to utilise external debuggers. This is probably not too much of a problem as the code is built on top of *gdb*, which is a popular enough choice amongst Linux developers anyway. Compilation however can be performed with any number of compilers simply by specifying a new command line to execute the compiler. The *Anjuta* developers envisage people using the software for creating more than just C/C++ programmers, so compiler flexibility is a must.

Despite coming across as slightly unfinished, it is clear that *Anjuta* offers the core feature-set that all development environments should. The question remains though – how does it fare for day-to-day usage? The answer – quite well actually. The application runs pretty smoothly, and there were no crashes to report during usage. It was able to import external projects fairly well, and has a useful plug-in system that can be used to extend itself (a useful class creation plugin and a patch plugin were supplied with the build reviewed here).

Some reasonable documentation is included with *Anjuta* which is informative and useful, but like the rest of the application feels very much like a work in progress. It is a very nice application though – literally, too, it even asks if you found the bug when you stop the debugger! It might not be the best development studio there is at the minute but this has the potential to go a long way. It wouldn't do any harm to widen the support to a more general audience rather than the GNOME/GTK bias there currently is though, and make a few last finishing touches to "professionalise" it for the full 1.0 release.

STOP PRESS: As we went to press version 1.0 attained full release, and the application continues to improve.

However, as we noted in the review, there are still a few rough edges – so why not contribute some help? The website notes a need for developers to help with new code, as well as bug fixes. There is also a request for help with the growing documentation – including the useful online manuals (currently in English and Japanese), as well as articles on the website.

LINUX Format VERDICT

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

An extremely good application for something so young – with the right finishing it will be superb.

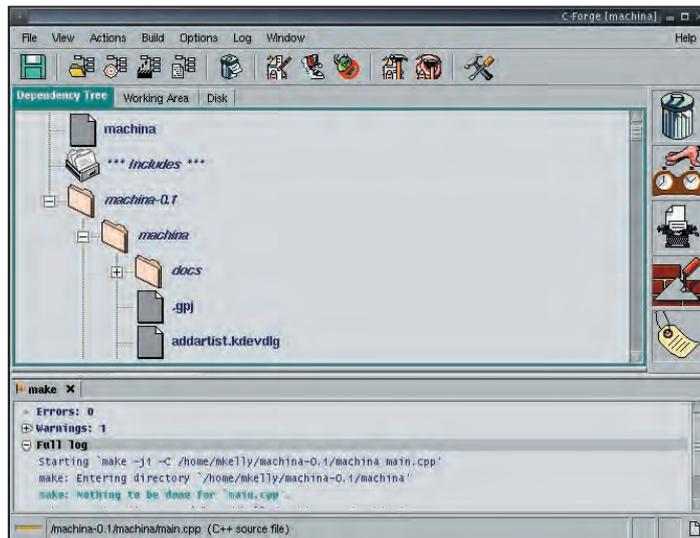
LINUX Format RATING

8/10

Code Forge

Feature-packed *Motif*-based IDE.

- **VERSION** 2.6e-2.2 ■ **WEB** www.codeforge.com
- **PRICE** from US\$40 – see text



Managing the files within your project is managed from here – confusing at first but comes with time.

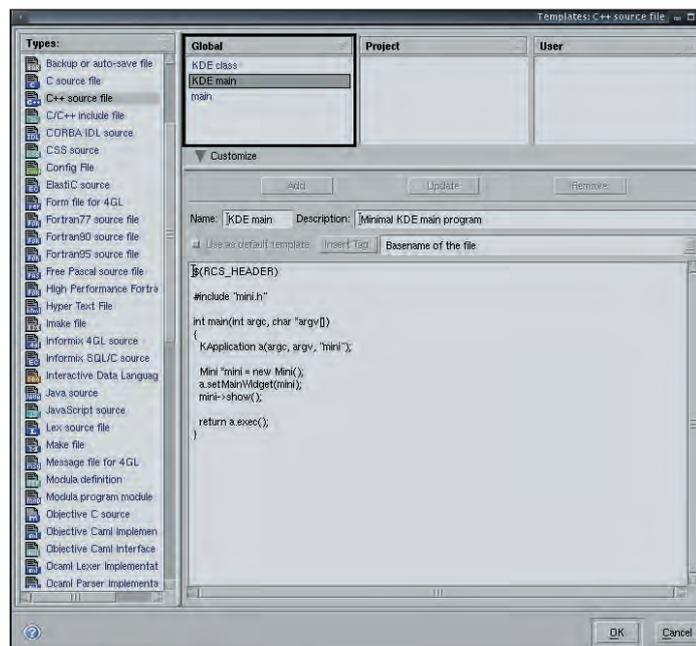
Code Forge (or C-Forge depending which part of the website/docs you read) is older than most of its competitors. It was developed as a unifying in-house tool by its parent company Code Forge Inc., who subsequently decided to further development, and release it to the public in 1998. The current development version will eventually be released as v3.0 but the current version available to the public is v2.6 which comes in three main flavours.

The first is the commercial edition which comes with the full range of tools and no deployment conditions, but at a price – \$40 for students, \$75 for private use and \$300 for commercial usage. The second is the free edition which comes with a limited set of tools, a restriction on developing commercial applications and no price tag. Finally, an evaluation version is available time-limited to 45 days (despite it saying 30 on the site.) It too stipulates no development of commercial apps but does come with the complete toolset. The evaluation version was used for this review, and was easily downloaded from the site as an RPM file. Downloads were available in deb format and RPMs for i386, Alpha and Power PC, and there were also Russian versions.

Installation from RPM was flawless. Upon first loading, the app offered the

opportunity to read up on the documentation which, while not a groundbreaking feature, is a nice idea. The documentation is good, especially the explanation of their project/target concepts, and is very commendable. All the dialog boxes I tried had a help icon that spawned context sensitive help.

The app is built using *Motif* and so



Code Forge has a large number of pre-defined source file templates, with the option to define as many more as you wish.

looks a little bit dated, which is unfortunate as the feature set is reasonably current. While functionality should be the king in this situation, when you spend a lot of time in one environment, it is important that it feels good, and at times *Code Forge* didn't. The docs were helpful, but it was still difficult to get into development and the *Code Forge* way of thinking. Like *KDE Studio*, it has a different way of managing projects, whereby you can specify a number of targets. These can be libraries, executables, etc., and can have shared dependencies which can also be targets. This is a powerful way of working, but it would help if there was some sort of guidance or pre-defined projects. The working layout is also different from the competition – the main window is the project manager. Once you create or open a project a new window is spawned to manage the targets for that particular project. It is good to have the option of working on multiple projects at once, but it could easily lead to confusion.

In actual operation though, *Code Forge* is actually quite good to use. The initial feelings of confusion as to where to actually put the source code were soon forgotten as I clicked around getting things set up. It certainly isn't the most intuitive of development environments – it took me a while to figure out where I could set up include paths and libraries to link against.

Once I found the location though, the wealth of configuration options was extremely impressive.

While there was no way to utilise your own editor, the built-in effort was very commendable. It came stocked with a wealth of options to control the way in which it worked. It was also fairly well integrated into the overall suite and had access to some nice tools such as a diff/merge program.

Code Forge comes with no integrated debugger at present. This is being rectified in v3.0 at present, but for now the only real debugging option is a field in which you can enter the name of the external debugger to use. While this is a reasonable solution, the concept of an integrated development environment demands somewhat more integration than this.

There are some useful features, such as class diagrams for C++, support for revision control systems and a great system for managing source code templates (with a wide range pre-defined.) It supports options for a wide range of compilers and, although not really relevant in this roundup, has a wide range of language support. This package really does feature an awful lot for the price you pay. \$75 dollars is a small price to pay, although if you can live without features such as revision control, a limitation on the number of open windows and support for languages other than C/C++, then you may want to take a long hard look at the free version instead.

If you can get your head round the project/target system, you could find yourself becoming very productive with this IDE. The commercial version has a lot going for it, but in my opinion not too many people will be tempted to part with the cash for the extra features in light of the what they can get for free.

LINUX Format VERDICT

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

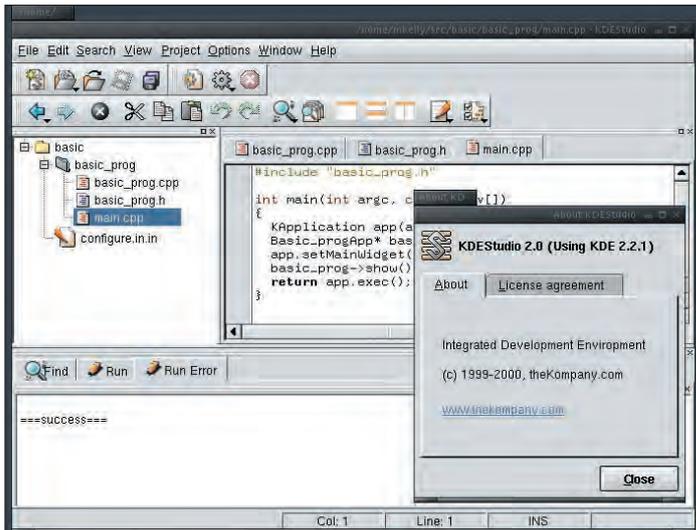
A great piece piece of software that could really cope with more complex projects (but could do with a makeover!)

LINUX Format **RATING**
8/10

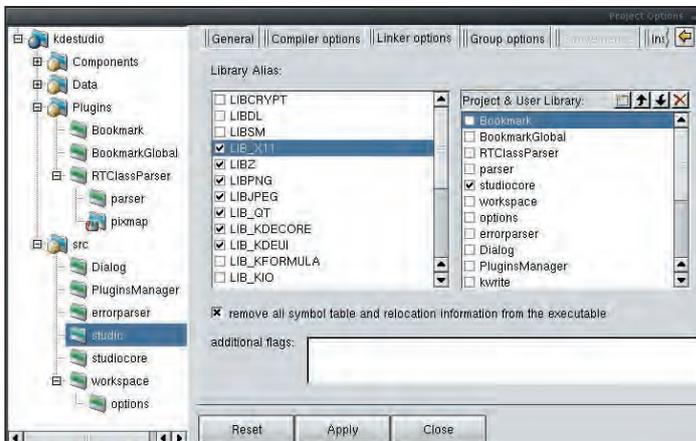
KDE Studio Gold

TheKompany's commercial fork of *KDE Studio*.

■ **VERSION** 3.0 ■ **WEB** www.thekompany.com ■ **PRICE** US\$24.95



A free version of *KDE Studio 2.0* is still available if you know where to look.



The **Workspace/Project** concept is tricky at first but does allow for finer project control.

KDE Studio Gold is a commercial

IDE released by the ominously named "The Kompany." Like another of their products – *Quanta+* the HTML editor – *KDE Studio* actually comes in two forms, *KDE Studio* and *KDE Studio Gold*. *KDE Studio* was forked in the past and the commercial interest went into the *Gold* codeline. The original *KDE Studio* appears to have been left to languish in obscurity. It was originally going to be used for this review, but it turned out to be a bit of a treasure hunt just to find the source code (hint: Google search for "kdestudio-2.0.0.tar.gz"). Don't be fooled by the website linking to the source and binaries on their FTP site,

actually going there is a fruitless task.

As it is a commercial application the *Gold* version will cost you money to get hold of, although there is a demo version available for download from the site. Beware though – you don't get the source, and compiled versions of the demo are provided only for Red Hat 7.2, Mandrake 8.1, Slackware 8.0 and SuSE 7.3. The demo is the full version but is restricted to use for 15 minute periods only. The full application also comes in two flavours, *Standard* and *Professional* – for \$25 extra you'll get context-sensitive documentation on *Qt/KDE/lib/kernel* documentation. That really appears to be the only difference.

In operation *KDE Studio* appears to be very much in the same vein as *KDevelop* and *Anjuta* in that the general layout is a status window at the bottom, editor panel at the top right and a sidebar at the top left. This seems to work fairly well for most applications, and like the rest, *KDE Studio* is capable of floating the various panels (although not the toolbars.) A large number of status windows, toolbars and sidebar panels are available to show through the View menu, although it would be nice to have them logically grouped to avoid the need for experimentation just to find out what some things are. Worth mentioning though is the Class Diagram panel which, as the name suggests, has the capability to diagrammatically display class inheritance – this could be a useful aid to those developing new code, or trying to understand somebody else's!

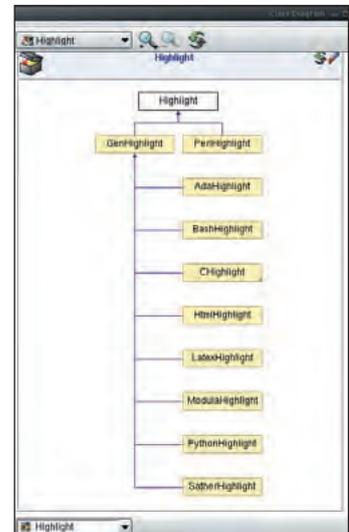
The editor is built-in and is a little bit poor compared to some of the other offerings. It does support tricks like code "folding" and syntax highlighting, but the range of options available and the extent of the syntax highlighting is limited compared to those supplied by the competition. The application itself has an extremely poor set of options with very little left to the user to configure – I still haven't discovered if I can change the compiler in use.

Unlike some of the competitors, *KDE Studio* has a somewhat strange way of managing projects. Upon starting for the first time you will be asked if you want to create a new workspace. Whilst at first this may seem simply like a terminology change, you soon realise that within your workspace you can have more than one project. Workspaces can be of a number of types such as KDE, Qt, X or command line, while the projects within can be shared libraries, executables and data structures. You can then drop in individual files to the projects (such as KDE main, Qt main, source files, and header files.) It does make sense, and is nice for managing larger projects (as the building of individual portions can be independently managed,) but is initially confusing and may be more than is necessary for many users.

Unfortunately there was no documentation included with the demo version and I was unable to determine what level of information

was provided with the full version. It would have been nice to get some more information on the debugger – it seemed to work okay, but it wasn't very responsive. All the usual debugging options were there (stack trace, watch variables, etc) but annoying little niggles such as not being able to set a breakpoint by clicking in the gutter or right clicking on the code meant that the debugging experience was not as fluid as it should be and it would have been nice to be able to select your own debugger to use.

Considering that the main competitor for *KDE Studio* is *KDevelop*, I would have a hard time recommending that anyone pay for this product. It is fair to say that were the full version available for free I would still not recommend it. Other packages provide far more in the way of flexibility and provide a much better user experience and for a commercial application it just didn't come across as professional enough for my liking.



Every cloud has a silver lining – the class diagram within *KDE Studio Gold* is an excellent addition and something the competition should be working on.

LINUX Format VERDICT

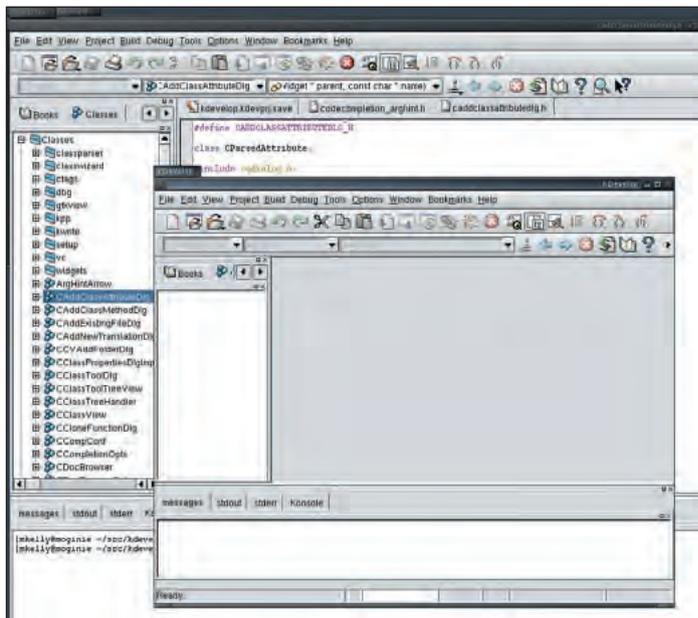
Installation	6/10
Documentation	-/10
Features	8/10
Ease of use	6/10

Not a terrible application but simply outshone by some of the others on test here.

LINUX Format RATING

6/10

Roundup C++ IDEs



Which came first... it is possible to load the source and compile the application from within itself.

KDevelop

KDE's polished development environment.

■ **VERSION** 2.1.2 ■ **WEB** www.kdevelop.org ■ **PRICE** Free

I won't hide the fact that I like *KDevelop*, and it's really fair to say that I like it a lot. Any program that is this comprehensive and easy to get to grips with is going to get praise heaped upon it. The fact that it is a non-commercial effort is a huge plus point and this kind of quality software doesn't often come so cheaply. The project was started in 1998 and has rapidly developed into a remarkably slick application.

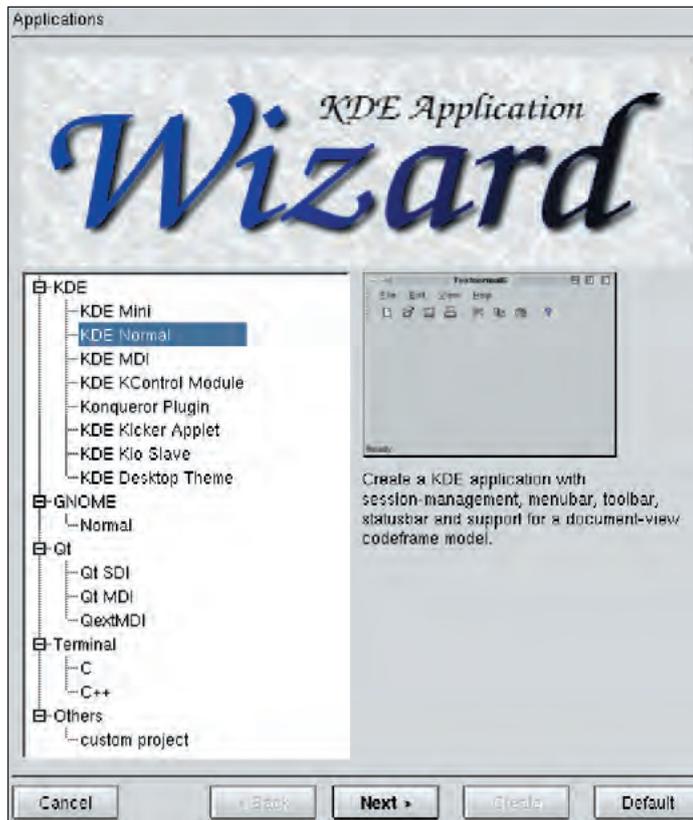
Running it for the first time spawns a setup wizard which allows the user to make some preliminary choices about editor and document layout, and picking up the the tools and docs available on the system. The main application then appears with a nice blank canvas to start creating with. Some people would prefer quickstart wizards, but hardcore programmers tend to prefer to get stuck straight into what they want to do.

While it is possible to simply hit the File menu, open a single source code file and begin editing straight away, what you'd be really missing about this app is it's excellent project features. The *KDE Application Wizard* is a couple of menu clicks away, and offers a wealth of options to get you started. As indicated by the name of the wizard there is a distinct bias towards KDE-based apps, although this is understandable given

that *KDevelop* runs under KDE. There is the facility to create GNOME, Qt and custom C/C++ based apps, although support is not quite as good as for straight-up KDE stuff. When you've got a project created it is easy to manage it through the project menu which allows for file and class management, docs and source tarball creation, as well as compiler and linker options.

The standard *KDevelop* working environment consists of a Multiple Document Interface (MDI) which shares app space with an output window and a tree view. The tree view offers a number of ways to view the project including raw files, a class browser, documentation and a file groups tab, which groups the project files into useful categories such as header and source files.

Most of the screen is devoted to the editor, a version of *KWrite* built into the system. The editor is reasonably good, with plenty of the features demanded by developers (such as configurable syntax highlighting, auto-indenting and tab-stop replacement.) Unfortunately the programming related options available to other editors far exceeds what this one is capable of, and there is no way to use your own favourite editor instead. It offers *Emacs*-style keyboardings, but that doesn't help us poor *Vi* users!



The *KDE Application Wizard* boast a variety of pre-determined projects to get you up and running with a minimum of fuss.

In its favour the editor is nicely integrated with the built-in debugger and you can easily set breakpoints visually in the same window that you are editing your code within. It would have been nice to have a separate floating toolbar containing the debugger commands but you can't have everything you want. It is possible to use external debuggers, although as usual, it is considerably easier to work with the built-in which has a reasonable enough range of options to satisfy most people.

In general use *KDevelop* is an extremely nice app – it looks good, and whilst sometimes a little cluttered, has plenty of options available from the main toolbar. A single click is all it takes to compile and run the entire application. Compilation takes place using a choice of compilers (*gcc* by default) and all output is handily dumped into a message window at the bottom of the work area.

While it serves as a general C++ (and C) editing environment, this app really is best geared towards KDE/Qt development. It has plenty of docs available within the work area for both technologies (although this is dependent on you having said docs pre-installed on your system.) Older

versions of *KDevelop* had a built-in KDE widget layout tool, but more recent versions depend on you having pre-installed Troll-Tech's own solution *Qt Designer*. This is accessed via the Tools menu item, which is configurable and allows you to add handy links to your favourite helper applications.

There is no denying that this is a top quality app. Recent versions support the new KDE3 and the developers do a great job of keeping abreast with KDE development. There's plenty of scope for using this program from beginners to hardcore developer. It has something for just about anyone except the most demanding of users, and when this functionality all comes for free it's really very difficult to complain.

LINUX Format VERDICT

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

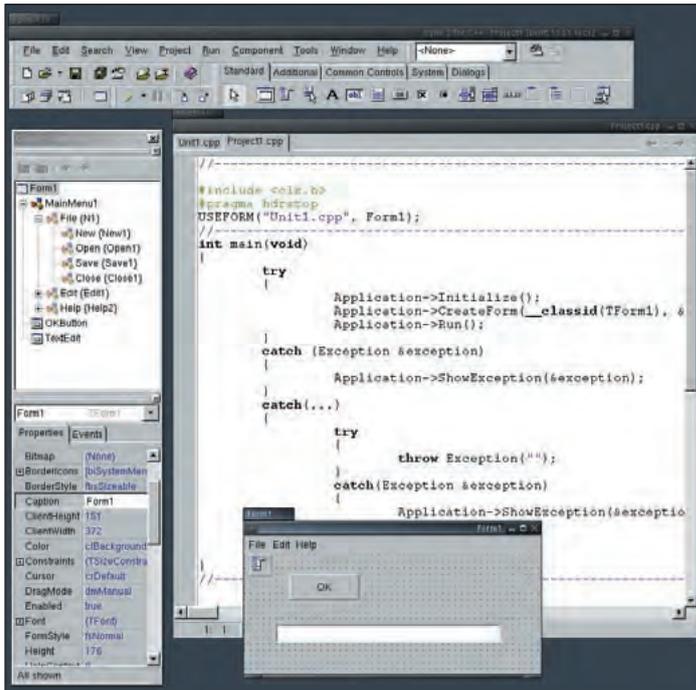
If you want to develop for KDE this is the one – a testament to what open source projects can achieve.

LINUX Format **RATING**
9/10

Kylix Open Edition

Linux port of Borland's *C++ Builder*.

■ **VERSION 3** ■ **WEB** www.borland.com ■ **PRICE** Free



Full *Kylix* environment can be a little odd to get used to but works in time.

"Hold on just a second," I hear you cry, "this is supposed to be C++ IDEs!" Yes, hands held high, I confess that *Kylix* is *Delphi* for Linux, but you may not realise that it's effectively got Borland's *C++ Builder* in it these days. *C++ Builder* is an immensely popular developer tool on the Windows platform, and it was only a matter of time before it hit the Linux world. It has done so as part of the fast-growing *Kylix* package, which began exciting Linux users just a few years ago. Now on its third outing, *Kylix* can be started up in C++ mode as opposed to *Delphi* mode (using `startbcb` at the command line instead of `startdelphi`.)

Getting hold of a copy of *Kylix* is relatively painless, simply head along to www.borland.com, hit the downloads section, and look for the *Kylix 3 Open Edition*. It is free to download and use, although it does have licensing restrictions and isn't as fully featured as the editions that you can pay for. The only real drawback is that the install file is quite large at over 95MB (but if you have LXF 33 look on the coverdisc). In order to run it you will need a 2.2 or

higher kernel and *glibc* 2.1.2 or higher. Installation was painless on a Mandrake 8.1 box. You will need to register with the Borland Developer network in order to get a licence key, which should be copied into your home directory so that the application will start up.

If you are used to developing apps with Borland's RAD tools then you will be instantly at home with *Kylix*. The interface is pleasingly familiar (although I did have initial problems with the individual application windows not integrating too well with the sloppy focus method employed by the *Fluxbox* window manager. It does appear to be the Windows application running a version of *Wine*, and so at times the interface looks somewhat inconsistent, but it doesn't form much of a problem.

Kylix has features that allow you to completely manage your project from within the app. The components used (compiler, linker and debugger) are all Borland's own – so there isn't much scope for using your own preferred tools. The upside of this is that the tools are designed to run together and so there should be little or no problems in usage. Similarly the editor is built-in

although it does have many options that you can tweak to your satisfaction.

The work environment consists of a panel that floats along the top of your screen containing all the toolbars you need to work with, although they are detachable. Also floating as independent units are the object tree, and object inspector palettes. The tree allows you to inspect the application hierarchy while the inspector allows for individual attribute editing. When compared to creating a Qt application directly in source code, the inspector is very handy, but doesn't allow for the same level of intimacy with the code.

The remainder of the standard layout is taken up with a form window and the text editor. Again these both float independently (although the object inspector and tree can both be docked in the editor) which leads to a rather disjointed app. Still, Borland have been using this setup for a long time now, so it must work for many people.

The final applications are based around a customised version of the Qt toolkit. Qt is the toolkit behind KDE and so your applications should fit reasonably well into many Linux desktops. Building your application is simple, and they can be easily distributed if packaged with the appropriate runtime libraries.

Applications developed with the *Open Edition* must be distributed under the

GPL, so you'll need to cough up if you want to create proprietary apps.

While it is a great application *Kylix* does have a limited audience. If you don't really want to develop GUI applications and simply want something to manage your latest C++ project, then you are out of luck. And while this is C++, there is something of a learning curve to using *Kylix* – for the most part this is point and click development and if you are not used to Borland's way of doing things, you are going to have to get used to it or head elsewhere.

Considering that this tool provides for free what you have to pay for with two Windows-based applications, there isn't really much to fault. While the RAD approach might seem ideal for novice programmers, it hides a lot from the user which could be a detriment when later moving to GUI programming at a lower level. As a result *Kylix* is probably best for experienced users who simply want to develop in a hurry.

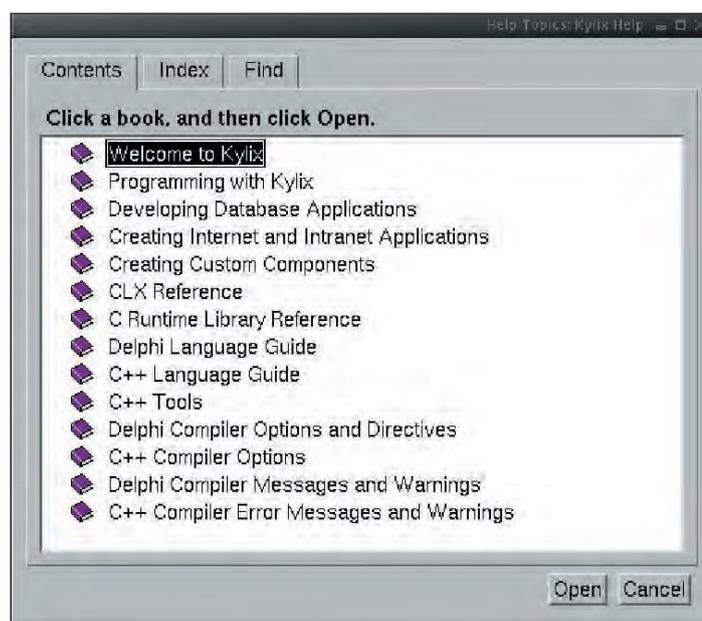
LINUX Format VERDICT

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

A fantastic package long overdue for Linux C++ programmers, but not quite flexible enough for top marks.

LINUX Format RATING

7/10



A wealth of options are available from the Help menu, and much of it is available as context-sensitive help from program dialogs.

Code Crusader

Proprietary *Code Warrior* clone.

■ V 3.0.3 ■ WEB www.newplanetsoftware.com/jcc ■ PRICE US\$84

According to the Freshmeat entry *Code Crusader* is inspired by *Code Warrior* by Metrowerks. Having never used *Code Warrior* I cannot verify this, but if you have then you know what to expect from *Code Crusader*. A 30-day time-limited demo is available from the New Planet Software website – other restrictions include 5 open file limit, only 5 files per project and a limited number of search paths per project. The full software is available for \$84 as part of the *JX Development Suite*.

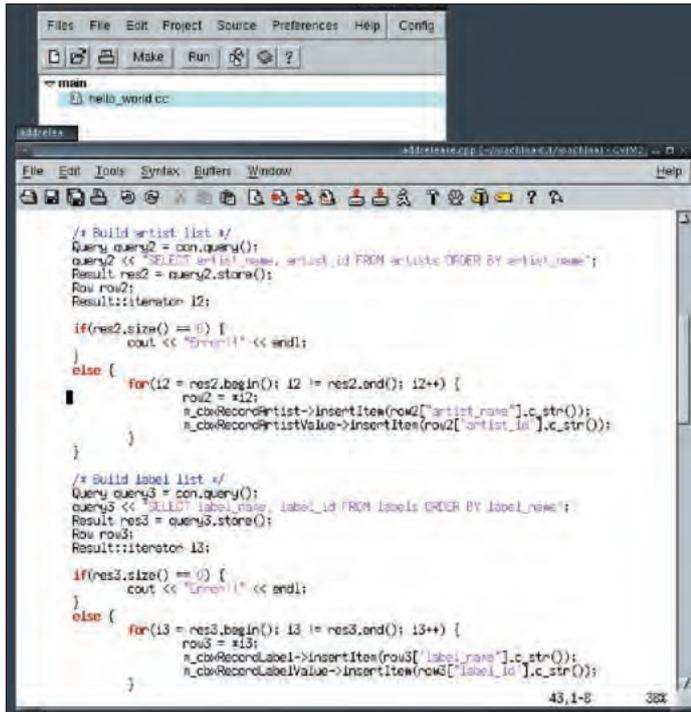
The application has a very basic feel to it, and you could be forgiven for not realising that this was a commercial offering. The level of configuration on offer is paltry compared to the competition, although I was impressed that I could choose to use an external editor rather than the built-in offering. The built-in editor was competent although uninspiring and unfortunately there were no real debugging facilities offered. The debugging option was to

include a path to your preferred debugger (defaulting to *Code Medic*, the company's own wrapper around *gdb*.)

Opting to create a new project presented you with a basic list of application types to create including C, C++, Java and Qt projects. Makefiles

were created for you automatically by using *make* or *trmake*, which worked fine for basic applications, but I had trouble trying to compile a simple Qt application. The only way to add the Qt lib and include paths for the compile process was by editing the *Make.header* file directly. This isn't a huge problem, but the fact that the other IDEs on test here give you nice config boxes to enter such data reflects very badly on *Code Crusader*.

Overall it is very fair to say that I was wholly unimpressed with this application. Its features and appearance are basic and dated, and there it does not add much to the development process other than make it easy to kick off the basic building blocks a developer might use at the command line.



It's unfortunate that the best feature of *Code Crusader* is the ability to utilise external editors.

LINUX Format VERDICT

Installation	5/10
Documentation	6/10
Features	4/10
Ease of use	3/10

An extremely disappointing addition to what appeared to be quite a strong roundup.

LINUX Format RATING
 4/10

QIDE

Problematic IDE, ported from Windows.

■ V 1.0.0 ■ WEB www.q-software-solutions.com/qide ■ PRICE €30

QIDE is a GTK-based application that is supposedly a port of *LCC-Win32* to Unix-like platforms, although for some reason there are a number of references to *Wedit* in the tutorial and help files. It comes at a price of 30 and there is a demo version available for download so you can try before you buy – which times out after a certain period, requiring a restart.

The application interface is extremely clean, featuring a menu-bar/toolbar, editor window, sidebar containing open files and a messages window at the bottom, with most functionality accessed through the menu. Creating a project is quick although the so-called application wizard is more a birthday party conjurer than a Gandalf. Once your project is created there are a wide variety of

options left to tweak, offering complete control over the development process.

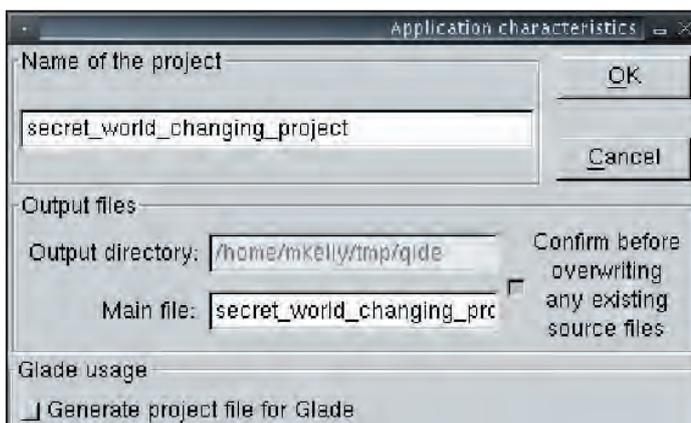
A range of tools (and wrappers for

external tools) is provided within *QIDE* including a built-in (but bog-standard) editor and a debugger which wouldn't execute without the app locking up – not an impressive demo! CVS support was a useful addition as was the ability to apply source code patches.

The biggest issue with this software was that with a basic Hello World app it was incapable of creating an appropriate

makefile straight off, yet many of the IDEs on test were capable of generating a respectable looking basic GUI app, but *QIDE* was incapable of creating a working makefile – an acknowledged bug which should be fixed in the future. First time users are presented with a basic tutorial, and a large amount of HTML help.

Overall a reasonable app, despite a few irritations. It's not for newbies – if you're not familiar with the mechanics of managing source code compilation then you would be better off elsewhere.



If this is what they call an application wizard then the magic has gone.

LINUX Format VERDICT

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

OK if you know what you're doing, and are prepared to get your hands dirty.

LINUX Format RATING
 5/10

C++ IDEs THE VERDICT

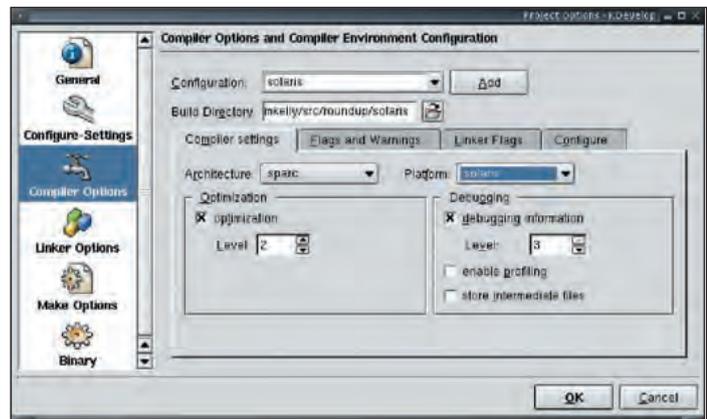
So what really makes a good IDE? The level of control, the fluidity with which the developer can churn out code, abstraction from the guts of project management? Unfortunately (for me, that is) it really is a mixture of many factors, and an awful lot of it is left to personal opinion. I really hate it when this happens!

Being Linux though I'm prepared to make a few assumptions about who the average Linux developer is, and what that average person wants from an IDE. The average developer is after an environment that gives a certain amount of flexibility, removes the need for mundane "housekeeping" tasks, and offers a wide range of tools to speed up the whole development process. This means that the IDE must

be highly configurable, fairly well automated, and laden with ways of doing things quicker.

In the respect of these factors there could only be one real winner in my opinion, and that was *KDevelop*. It could be said that there was some bias in this decision, as *KDevelop* is an application that I used in the past for a university project, but part of the remit for that project was to research the available IDEs and choose the best for the application to be developed. Back then I chose *KDevelop 1.2* and I have no hesitation in choosing *v2.1* today.

I first used it knowing nothing about C++ or Qt/KDE development but found it very easy to get to grips with. It still is, and with a wealth of documentation and an excellent user interface this application is going to be



KDevelop: A wide range of project options are available including the ability to add multiple target platforms for the compiler.

very tough to beat. A fantastic level of control over the project is offered, and support for a number of varying project types was more than welcome as it didn't preclude non KDE developers from taking advantage of the features on offer. I think what makes *KDevelop* so good is the fact that it brings together so many useful tools, both internal and external, into an environment which is a pleasure to use.

Basically *KDevelop* manages to make programming better by making the whole development process more fluid and a lot more intuitive. Not only this, but the project is extremely fast moving and a number of new features are being developed in the upcoming 3.0 version which is currently in alpha testing. These include support for languages other than C/C++, *qmake* support and C++ autocompletion. It is great that an application which is open source is such an amazing contender and manages to put some of the proprietary commercial offerings to shame. There is a lot to be learnt from *KDevelop*!

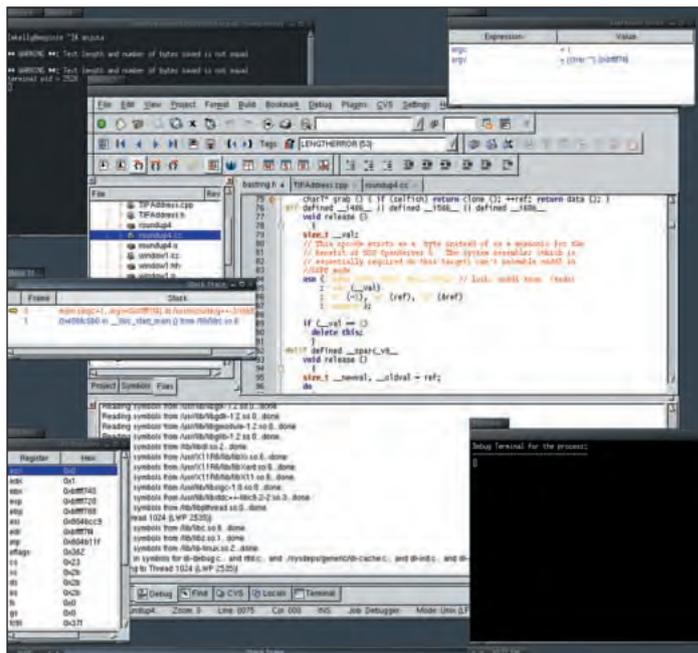
What of the competition? Despite being a great application and long overdue for Linux, I felt that *Kylix* was not an IDE for the average Linux developer.

I feel that *Kylix* is great in many respects but that it can be too abstract from the nuts and bolts of the development process to suit what most developers will want. Having said that, it will be interesting to see how *Kylix* fares in the future – it will encourage a swathe of new developers to use Linux, and it may soon alter the demands of the average developer.

Anjuta was a fantastic application, let down only by its youth. With a little bit more development time this will be quite a contender, and if they throw in more KDE/Qt support it will end up being quite a close call between *Anjuta* and *KDevelop*.

Code Forge too was a sterling performer despite my initial reservations – what at first seemed like a somewhat quirky way of working soon revealed an extremely powerful and flexible tool underneath. It just didn't have the snappiness or finesse of *KDevelop* in order to pip it to the post.

The great thing about this roundup is that it highlights the fact that there is a real variety of environments out there. From RAD solutions to middle of the road IDEs, to the hardcore *xterm*/editor/compiler brigade, there really is something for everyone. It is easy to write Linux off as an unintuitive hackers paradise, but this is really not the case any more. [LXF](#)



Anjuta: A debug session – it's not the most fully-featured debugger in the world, but is a pleasure to use.

Table of features

IDE	Free	Built-in Editor	Syntax Highlight	Code Folding	Debugger	Class browser	Project management	Application wizard
ANJUTA	Y	Y	Y	Y	gdb	Y	good	Y
CODE CRUSADER	Y	Y	Y	N	none	Y	poor	N
CODE FORGE	N	Y	Y	Y	none	Y	excellent	N
KDEVELOP	Y	Y	Y	N	internal	Y	good	Y
KDE STUDIO GOLD	N	Y	Y	Y	internal	Y	excellent	N
KYLIX OE	Y	Y	Y	N	internal	Y	good	N
QIDE	N	Y	Y	N	gdb	N	poor	N*

* Claims to have one but a very poor effort!

HotPicks

The best new open source software on the planet!



Richard Drummond

As well as writing our Java series, Rich finds time to try new Linux apps.

This is the place where we get to profile some of the hottest software around.

Each month we trawl through the hundreds of open source projects which are released or updated, and select the newest, most inventive and best for your perusal. Most of the Hot Picks are available on our coverdiscs, but we've provided web links if you want to make sure you have the very latest version.

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

HotPicks at a glance

Gringotts	44
gHack	45
Childsplay	45
flphoto	46
GraphThing	47
FIAIF	47
ReZound	48
GTK-Lsof	48

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!

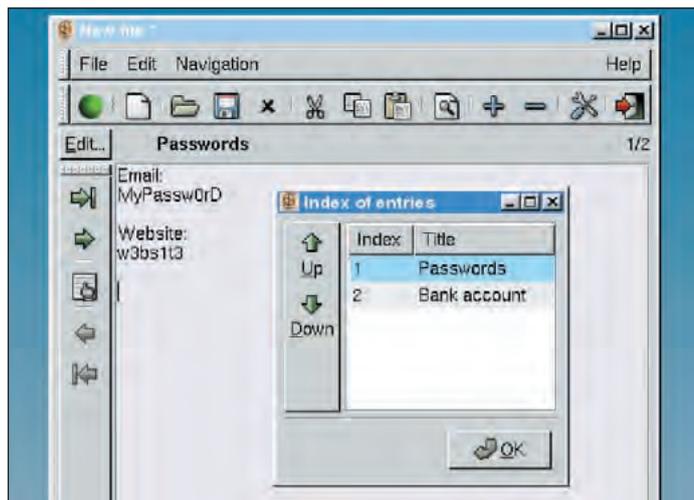


PERSONAL SECURITY TOOL

Gringotts

■ **VERSION** 1.1.1

■ **WEB** www.prosa.com/people/grizzo/gringotts/index.php



Gringotts lets you enter pages of text and store them away safely.

Being one of the few – it would seem – not to have read *Harry Potter*, the reference to *Gringotts*, the impregnable Wizards' bank from the books, was completely lost on me. Nevertheless, it is an apt name for a tool which creates a virtual safety-deposit box for storing your sensitive, personal information, like passwords and credit card details. Which is just what this *Gringotts* is.

Stripped down to basics, *Gringotts* is simply a text-editor which saves text in a compressed and encrypted form. Why not just use *Emacs* and *GnuPG*, you ask? Well, because *Gringotts* integrates the whole process and makes it quicker and easier. *Gringotts*' swish GTK2-based interface lets you input and manage multiple pages of text. This allows you to, say, collect passwords on one page, bank account details on another, and so on. From here you can cut-and-paste text to wherever you want, for example, to input passwords into a dialog. Moreover, it gives you the choice of various block ciphers for encryption – such as AES, Twofish and 3-DES – and it can use

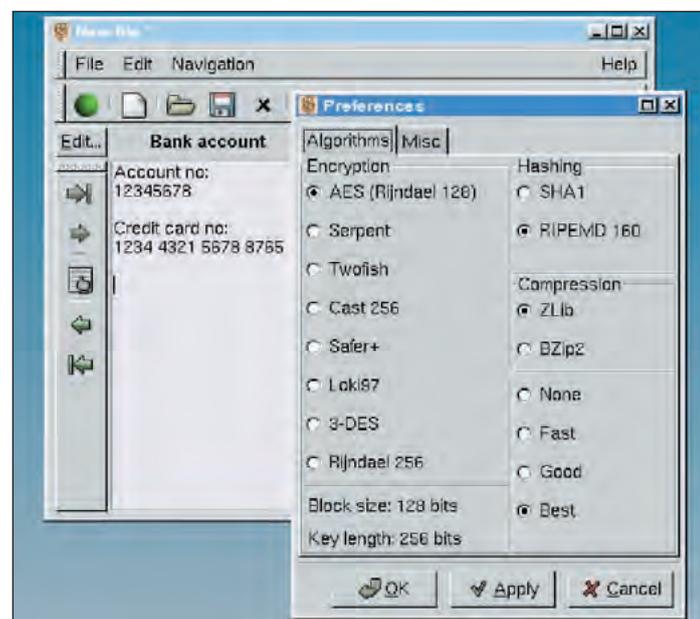
either SHA1 or RIPEMD 160 for hashing. *ZLib* or *BZip2* with varying compression levels can be used for compression.

Of course, any encryption system is only as good as the key you use to

encrypt, and *Gringotts* has a number of tricks up its sleeve here. You can choose a conventional password or passphrase, if you wish, and *Gringotts* will give a graphical representation of the quality of your pass-phrase as you enter it. Alternatively, you can pick the contents of an arbitrary file or write-protected floppy disk to be your key, and thus vastly improve your potential key size. Obviously, if you choose a file, then this must be a file that will not be modified or deleted at any point (similar *caveats* apply if you choose a floppy-based key) or you won't be able to access your encrypted data again. You should also be aware if you choose a file on a publicly accessible system, then you leave yourself open to a brute-force attack. An intruder could simply try all the files on the system. An obvious solution here is to store the file on a removable disk and take it with you. The documentation includes some more useful tips on how to using file-based keys.

Other neat touches abound.

Gringotts is a neat solution to the problem of how to store sensitive personal data. In an age where we are constantly being burdened with more passwords and codes than we can remember without writing them down, it's sure to find lots of users.



There's a choice of block ciphers, hash functions and compression methods.

RPG

glHack

■ VERSION 1.0 ■ WEB <http://glhack.sourceforge.net/>

Games don't come with a much better pedigree than *NetHack*. It can trace its roots back to what was arguably the first action game, *Rogue*. Games of this ilk have captivated players on dozens of platforms since the 1970s and will continue to do so. But to attract a new generation of gamers, *NetHack* and other *Rogue*-like games need to be made visually more appealing. Clearly, the ASCII graphics of yesteryear will no longer cut it. Thankfully, *NetHack* has evolved as it has acquired new platforms, and colourful graphics and mouse control have been added, as well as improved game-play.

glHack is a new port of *NetHack*, based on the latest *NetHack* 3.4.0 release. As the name suggests, *glHack* uses OpenGL for rendering. Don't get

too excited: it's still a 2D game. The use of *OpenGL* (via *LibSDL*) simply means that you get fast and crisp screen updates.

Unlike other recent ports, such as *GnomeHack*, *glHack* aims for the authentic experience. It eschews any widgets, menus or multi-window environments provided by modern GUI toolkits and instead presents the game in a single, optionally full-screen, window. Thus you get the maximum play area on screen, and all the user-interface eccentricities of the original are preserved.

glHack is very configurable, and you can choose screen size and the graphics or 'tile set' used for rendering the game. New in the 1.0 release is an isometric, *faux*-3D tile set. If you liked *Gauntlet*, you'll love this. (To enable



OpenGL rendering and a new 3D look add to the addictiveness and the level of immersion that *glHack* provides.

this mode, simply choose the 64-pixel-high tile set in the game options.) Of course, the original ASCII-based display-mode is still there if you are feeling particularly masochistic.

Although purists may disagree, *glHack* is one of the best current

implementations in the *Rogue* genre. The addition of *OpenGL* for speedy screen updates and the new-3D look really help to add atmosphere to what is already a deeply engrossing game.

I must confess to still being a *Moria* man myself, though.

EDUCATIONAL GAME

Childsplay

■ VERSION 0.44

■ WEB <http://childsplay.sourceforge.net/index.html>

Being the parent of an incredibly inquisitive two-year-old, who takes an equal amount of joy as her father, it seems, at bashing at a computer keyboard, and being a Linux advocate, I'm always on the look out for Free Software targeted at young children. While there are several open-source projects in the works that aim to provide younger users an introduction to computers, the quality of what's available does seem to lag behind the commercial software world.

A new offering that recently caught my eye is *Childsplay*. In terms of the experience it creates, it doesn't offer any more than much of the rest of the free educational software around, and is poorer than some.

However, what is interesting about *Childsplay* is that it is a

modular system, written in Python (using the *PyGame* bindings for *SDL*),

that provides a framework for creating software for kids.

The advantage of Python – over a lower level language such as C – is that it's ease-of-use should open up development to a wider audience, and thus make it possible for those whose expertise is education rather than programming to contribute. This is desperately what the open-

source educational market needs.

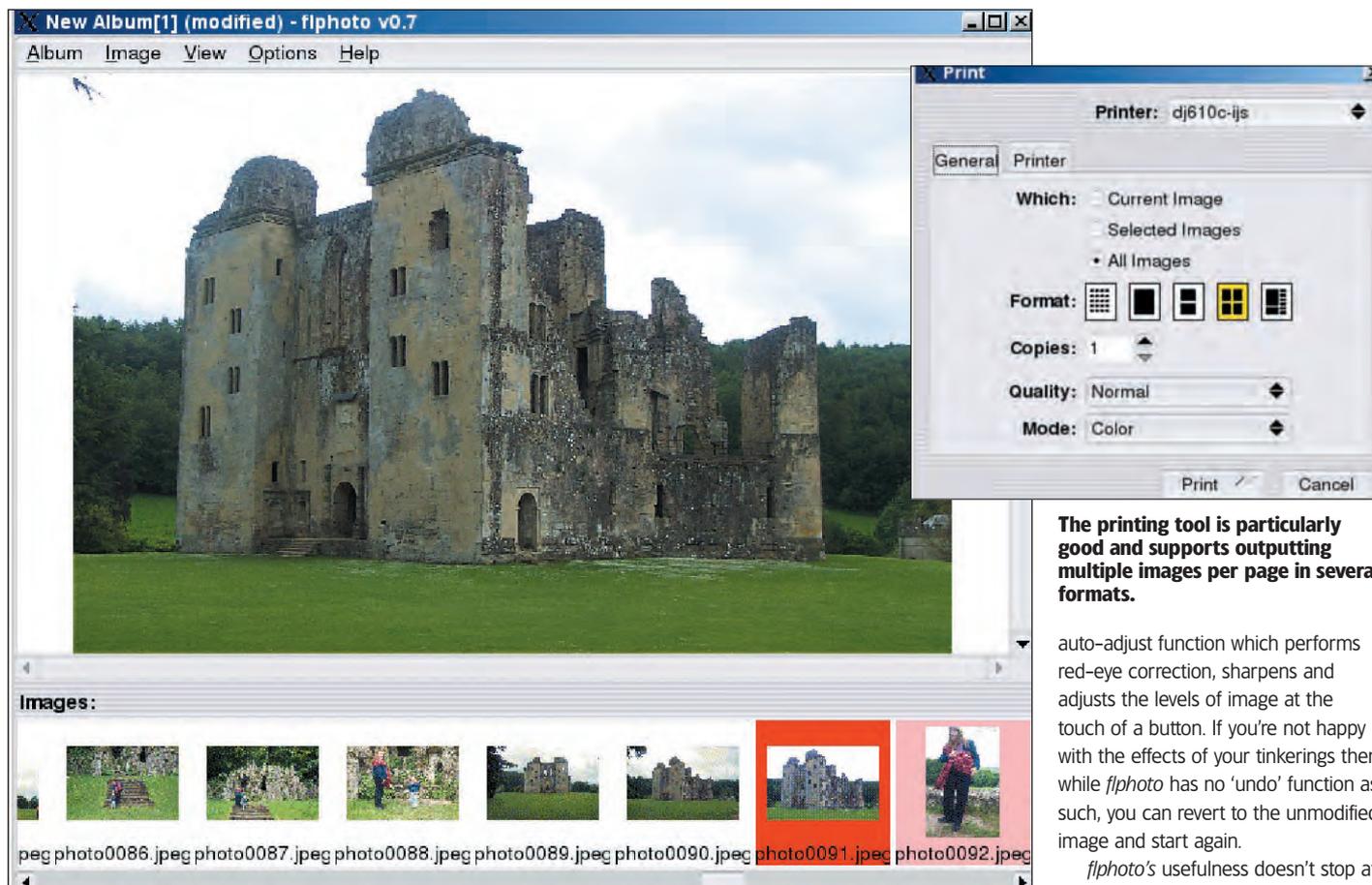
The games that *Childsplay* currently provides include a match-the-pairs-of-cards-type game, a simple arithmetic game and a word game in which the child has to type in the letters of a word as they appear on screen, as well as a game for toddlers which involves prodding animals with the mouse pointer to hear the sound it makes.

The graphics are colourful, the sound effects are cute, and the games target a range of pre-school ages. What *Childsplay* lacks is in game design. There needs to be more to hold a child's interest – more indication of progress, more variety, more rewards for getting it right.

Don't get me wrong. These are early days, and *Childsplay* is definitely a step in the right direction. I just hope more people get behind the project and develop it into something exciting. If you would like to help out yourself, then take a look at the webpage – where you will see that they need translators, proofreaders and people with ideas, as well as developers.



Childsplay is cute and colourful, but the project needs something extra to hold the interest of kids.



The printing tool is particularly good and supports outputting multiple images per page in several formats.

auto-adjust function which performs red-eye correction, sharpens and adjusts the levels of image at the touch of a button. If you're not happy with the effects of your tinkering then, while *flphoto* has no 'undo' function as such, you can revert to the unmodified image and start again.

flphoto's usefulness doesn't stop at manipulating photos; it can also present them. It features a slide-show mode, which will display the photos in an album, one-by-one, at full-screen size, and can print images. This works with any *CUPS* printer spool and provides a number of print formats. You can print one, two or four images per page, print an index of thumbnails, or print a 'portrait' collection, which fills a page with copies of one image at a selection of sizes. The 'print to file' option currently doesn't work, at least on my setup, and screws up the bounding-box and scaling of the image. I'd like to see more printer options, such as control over image placement and page margins, but *flphoto* currently provides one of the easiest ways of printing your digital photos, especially if you need to output multiple images per page.

flphoto is a long way from perfection, but, even as it currently stands, it is an indispensable piece of software. It fills a very obvious evolutionary niche in the Free Software ecosystem, falling between the all-singing and dancing image processors like *The GIMP* and simple image viewers. For people who need to simply and quickly manage and touch-up their digital photos, it's a god-send.

***flphoto* provides a fast and efficient toolkit for managing and touching-up your digital snapshots.**

IMAGE PROCESSOR

flphoto

■ **VERSION 0.7** ■ **WEB** www.easysw.com/~mike/flphoto

The mighty *GIMP* might be the pinnacle of image-processing prowess on the Linux platform, but for many of those simple, everyday tasks using *The GIMP* is like using a flame-thrower for lightly toasting a bagel: it's completely over-powered for the job. When you just need to crop, scale, or re-touch your digital snapshots, for instance, something lighter and more efficient is definitely needed. *flphoto* fits this bill quite nicely.

flphoto is a new image-processing tool designed specifically for handling digital photos. It's based on the *FLTK* widget (the *Fast Light Toolkit*) – which, as the name implies, is low-footprint and rather nippy, yet still powerful enough to create apps which are comfortable to use. Unsurprisingly, *flphoto's* interface is incredibly quick, and lets you perform the tasks you need with a minimum of

fuss. User-friendliness isn't sacrificed, though, and *flphoto* includes full on-line help (not that you should need it).

flphoto is new. At the time of writing v0.7 was available from CVS, which corrects some bugs from the previous snapshot release, version 0.6. If 0.7 isn't available by the time you read this, it's well worth checking it out of CVS. Despite its immaturity, *flphoto* is now reaching the point where it is actually useful. Most features slated for the stable release are now implemented.

flphoto can load and save images in most common digital photo formats, and it can understand EXIF tags in JPEG files. It can also interface with your digital camera via the *gphoto2* library, and so download images directly. Groups of images can be collected together as an 'album' and albums can be saved to disk, which is a great tool for helping you

organise your photo collection. All the images in the currently loaded album are displayed as thumbnails in a scroll-pane at the foot of the main window. Any modified images are outlined in red.

The currently-selected image is displayed in the main part of the window, and this can be zoomed, rotated, scaled and cropped as you might expect. The crop control isn't as easy to use as it might be, though. This pops up a thumbnail in which you can drag out an area with the mouse to crop to, but doing this on a thumbnail lacks accuracy. As well as these simple transformations, *flphoto* boasts a number of filters. It can sharpen, blur, and adjust the brightness and contrast of an image, and also includes a red-eye filter. All of these tools are one-click functions: they work automatically. The brightness and contrast filter, for example, builds an intensity histogram and uses this, effectively, for automatic gamma correction. You don't get the control over the results as if you'd set the black, white and grey levels yourself, but for most cases it works. For the ultimate in laziness, *flphoto* includes an



MATHEMATICAL TOOL

GraphThing

■ VERSION 0.7.1 ■ WEB <http://graph.seul.org/>

Mathematicians have some funny ideas. A straight line is a curve, and more or less any collection of points joined up by lines is a graph. Seriously, though, the graph is an important mathematical concept with many practical applications, especially in computing science. It is a useful visual tool for representing the relationships between any set of objects.

GraphThing is a neat little application that lets you play around with graphs. It allows you to construct any simple graph on screen (a simple graph is one that has no loops or multiple edges), optionally with labeled nodes and weighted edges, and lets you manipulate its layout by dragging nodes around with the mouse. It provides various tools to query a graph's properties, such as determining whether a graph is Eulerian, or whether it is connected. It can produce various statistics such as an adjacency matrix for a graph, its degree sequence, chromatic number and so on, and find the shortest path between two nodes.

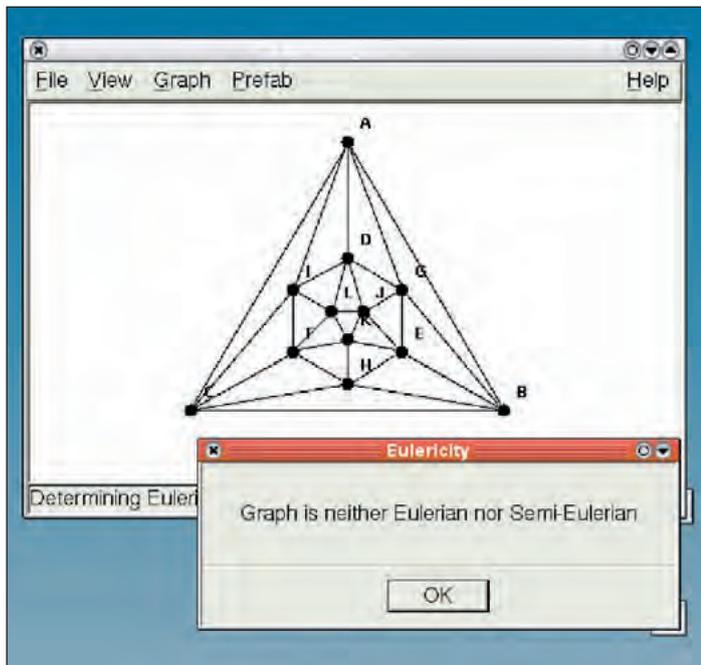
Although it useful as it stands, *GraphThing* can be improved. The way you create graphs at the moment is

rather limited, albeit incredibly simple to use. You must start with a 'prefabricated' graph with the desired number of nodes, and then add or remove edges to create the graph you want. You cannot later add or remove nodes to a graph.

So, for example, to construct a graph you might start with a null graph with the correct number of nodes and add the edges you need, or start with a complete graph and remove the edges you don't. *GraphThing* features a wide range of built-in graph types, such as complete graphs, complete bipartite graphs, cycles, wheels, stars, the graphs associated with the platonic solids and the Petersen graph.

Functions

As far as editing functions go, I would like to see *GraphThing* extended so that you can add arbitrary nodes to a graph. Support for directed graphs would also be useful, and the inclusion of some tools to manipulate the view, such as scaling and rotation functions. It would also be great if *GraphThing* would let you have more than one graph open at a time. This would allow the user to



Useful mathematical tool or just a join-the-dots puzzle? You decide.

visually compare graphs more easily, and then a function for checking whether graphs are isomorphic could be included.

Export

Better export tools are also a must. At the moment you can output graphs as a PostScript page, but this should be extended to include EPS and SVG formats as well, to make importing graphs into a document layout application easier. Some means of exporting a graph's adjacency matrix

would be a valuable addition. Although *GraphThing* uses a plain text file for storing graphs, the ability to export graphs in a more readily usable format would be useful. Perhaps a function for generating C source code with the graph represented as a static array could be added.

GraphThing has a lot of scope for development, but even now it will be a life-saver to the student of Graph Theory and will potentially save reams of doodling paper and much hair-pulling.

FIREWALL BUILDER

FIAIF

■ VERSION 1.3.1-1 ■ WEB <http://fiaif.fugmann.dhs.org/>

With today's concerns about network security, there is a plethora of firewall options to choose from. We have reviewed many in the pages of *Linux Format*, including firewall distros such as Smoothwall and dedicated appliances such as SnapGear's products. But, if you don't have the spare machine to dedicate as a firewall and don't want to fork out on a firewall appliance, what options are available? *FIAIF*, standing recursively for *FIAIF Is An Intelligent Firewall*, is one of the many firewall builder scripts out there which can simplify the setup of the Linux kernel's

internal firewall. This approach is much quicker and less error-prone than rolling your own firewall rules, and will work with just about any Linux distro. It is particularly useful if your firewall host is a dual-purpose machine, perhaps doubling as a desktop machine on a small LAN.

FIAIF is one of the more powerful firewall-building scripts available, being applicable to any LAN with a single Internet connection. It is written as a *Bash* script, so there are no particular installation requirements other than the usual networking tools

you find on every Linux box. *FIAIF* supports multiple network interfaces, notionally creating one or more network zones per interface. It is configured by a master config file containing global options, and then each named zone is configured by its own config file. The various configuration options are well-documented in the man pages accompanying the software.

The master *FIAIF* configuration includes options for setting up zones, setting up logging, and for loading specific *iptables* kernel modules. A zone's network parameters can be auto-configured from its interface's network connection or specified in the config file. DHCP is supported on the external interface, so dial-up and broadband connections may be used. Each zone configuration sets up the rules for that zone, and masquerading,

SNAT, port-forwarding and traffic-shaping is supported per zone. You can ban specific MAC addresses from a zone, watch traffic from a specific IP, and limit certain packet types (to avoid Denial of Service attacks). The default *FIAIF* configuration sets-up three zones, the usual external, internal and DMZ networks, with a sensible security policy. Thus the external zone accepts DNS, SSH, HTTP and HTTPS connections and limits pings, and the internal zone is masqueraded, and denies connections from other zones. This serves as a useful basis for most configurations.

FIAIF can take a lot of the stress of building your own firewall. It isn't as simple to use as the GUI firewall builder tools, but for those who need to retain the flexibility of an entirely custom solution, it is a useful and effective compromise.

AUDIO PROCESSOR

ReZound

■ VERSION 0.5.1 beta ■ WEB <http://rezound.sourceforge.net/>

Linux is fast becoming a viable platform for audio work. With kernel projects such as *ALSA* (the *Advanced Linux Sound Architecture*) and the work that has done to improve system latencies, such as the Pre-emptive scheduler, the technical reasons against Linux as a multimedia platform are steadily being chipped away. All we need know is the application software, and *ReZound* seems like a step in the right direction.

ReZound is an audio processor. There are several similar projects available, but *ReZound* is one of the most promising, combining some powerful features with a pleasant and easy-to-use GUI. It is built on the *FOX* GUI toolkit, and thus is lighter and more nimble than some apps built on KDE or GNOME infrastructure. I particularly like the dial controls that *ReZound* uses; they somehow seem appropriate for an audio tool.

ReZound supports the usual array of sound formats, as well as its own, and can record from any audio input device on your system. It uses *OSS* by default for sound I/O on Linux, but optionally can be built to use the *PortAudio* library (www.portaudio.com). This provides a wrapper for a host of different sound systems on various platforms, and supports both *OSS* and *ALSA* on Linux.

As well as simple manipulation of sound streams – such as cutting and pasting, overwriting, looping, etc. – *ReZound* includes a powerful set of effects, filters and remastering tools. This include familiar effects such as distortion, gain, and flanging and the usual range of filters. *ReZound's* secret weapon, though, is its support for convolution filters. This technique allows you to apply the audio characteristics of one sound to another, and is particularly useful for obtaining realistic reverberation. For



ReZound boasts a slick interface and some powerful audio processing tools.

example, you can record an arbitrary sound in a large room, and use that as the basis for a convolution filter which you can apply to your target waveform to make it sound as if it had been recorded in the same environment. *ReZound's* implementation is based on the *FFT* (*Fast Fourier Transform*) library, and is impressively quick when used

with a version of *FFT* built for your particular CPU architecture.

ReZound is still at a beta stage of development, but is perfectly usable. There's currently only rudimentary documentation, so finding your way around may prove difficult. Experimentation is the key, especially with the convolution filters.

DIAGNOSTIC TOOL

GTK-Lsof

■ VERSION 0.9.6 ■ WEB <http://digilander.libero.it/blaad/>

Lsof, or *List Open Files*, is one of the most massively useful tools a system administrator can install on a Unix box. *Lsof* does exactly what it says on the tin: it lists any files that are open on your system. But, since in Unix (and particularly in Linux) everything is a file, *Lsof* gives you the power not just to see what conventional files are open for reading and writing by which processes, but also lists what a process's current directory is, what devices it is accessing, what network sockets it is communicating over, and so on.

The problem with *Lsof* is that its command-line is complex and its man page rather dense. *GTK-Lsof* solves this problem, at least partially, by providing a *GTK+* 2 front-end. Its main window contains a tab-pane with three pages. The first of these displays the results of an *Lsof* query in a list view, and the second gives you controls to configure what information is listed there for

each open file, such as the command that owns the file, its process ID, the type of file, the file path and so on. The display can be set to automatically

refresh itself at a chosen interval, and you can export the list to a file.

The third-page of the main window lets you configure the *Lsof* query, and this is where *GTK-Lsof* falls down. It contains a list of switches that *Lsof* understands, and you can build-up an *Lsof* command line by picking switches or by entering the arguments directly into the text widget provided here. This allows you to

control the full power of *Lsof*, but doesn't work well in practice. You need to know what the various switches mean, because no help is provided. Even if the GUI just used more meaningful names for the *Lsof* options here, such as 'Process ID' rather than **-p**, it would be quicker and simpler to use. Despite this, *GTK-Lsof's* ability to present *Lsof's* output in a GUI make it a useful tool to have around.

COMMAND	PID	LOGIN	FD	TYPE	DEVICE	SIZE	NODE	NAME
kdeinit	856	evilrich	mem	REG	0x306	31252	225172	/usr/X11R6/lib/libSM.so.6.0
kdeinit	856	evilrich	mem	REG	0x306	79012	225171	/usr/X11R6/lib/libICE.so.6.3
kdeinit	856	evilrich	mem	REG	0x305	85695	20323	/lib/libpthread-0.10.so
kdeinit	856	evilrich	mem	REG	0x306	17068	225183	/usr/X11R6/lib/libXrender.so.1.1
kdeinit	856	evilrich	mem	REG	0x305	7424	20320	/lib/libutil-2.3.1.so
kdeinit	856	evilrich	mem	REG	0x306	55372	35497	/usr/lib/libz.so.1.1.4
kdeinit	856	evilrich	mem	REG	0x306	33140	33220	/usr/lib/libfam.so.0.0.0
kdeinit	856	evilrich	mem	REG	0x306	672344	4892	/usr/lib/libstdc++.so.5.0.1
kdeinit	856	evilrich	mem	REG	0x305	29632	6868	/lib/libgcc_s.so.1
kdeinit	856	evilrich	mem	REG	0x305	130964	19827	/lib/libm-2.3.1.so
kdeinit	856	evilrich	mem	REG	0x305	1109900	19282	/lib/libc-2.3.1.so
kdeinit	856	evilrich	mem	REG	0x306	78288	1014	/usr/lib/libaudio.so.2.2
kdeinit	856	evilrich	mem	REG	0x306	296900	225184	/usr/X11R6/lib/libXt.so.6.0
kdeinit	856	evilrich	mem	REG	0x306	269140	7253	/usr/lib/libmng.so.1.0.0
kdeinit	856	evilrich	mem	REG	0x306	198908	5488	/usr/lib/libjpeg.so.62.0.0

GTK-Lsof gives you the power to snoop over every file access on your system.

GadgetsGalore



GADGETS GALORE!

The Linux Format team, with a little help from Santa, take a look at eleven cutting-edge Christmas gadgets.

IRIVER IDP100 DATAPLAY MP3 PLAYER

■ £350 ■ www.mp3players.co.uk

You may not feel that the world needs more data storage formats, but the international technology consortium behind this new contender think otherwise.

Dubbed Dataplay, the 250 and 500MB discs are even smaller than a MiniDisc, and the big buzz is that they're going to be extremely cheap to make, and – as a result – cheap to buy. They're also shockproof, which should make them perfect for portable music players like this, the UK's first Dataplay device, the iRiver IDP100. Discs are also, at the

moment, write-once, which hasn't been trumpeted quite so loudly.

iRiver's player is a bizarre-looking beast, its yellow doughnut body encased by two protective slithers of plastic to protect it from knocks and scratches while out and about. Sonics are great, and it also plays AAC music files as well as MP3s, saving you space as well as offering better sound quality.

❄️❄️❄️❄️❄️ 4/5



FUJIFILM @XIA SLIMSHOT

■ £380 ■ www.fuji.co.uk

Bond gadget aficionados will love this one. The right size to fit

in your wallet alongside more prosaic credit and loyalty cards (it's a mere 6mm thin), FujiFilm's Slimshot is about as close to one of Q's toys as you can get for under £100. Don't expect snaps that you'll be able to print and make posters from, but you do get a 0.31 Megapixel CMOS sensor, which makes for VGA-quality 640 x 480 images. On high-quality VGA mode, you can fit just 26 shots onboard, but drop to the economy mode (320 x 240) and you'll squeeze in 101 shots at a quality that's only really good for emailing. The Slimshot's trump card over its miniature rivals is Autobrite, a clever bit of technology that adjusts light levels automatically, making the Slimshot good for frivolous snaps inside as well as out.



❄️❄️❄️❄️❄️ 3/5

FRIDGEPLAY

■ £9.99 ■ www.fridgeplay.com

While the idea of attaching a magnetic game-board to

a fridge might seem like a cunning marketing gambit, we reckon the originators of this series of games have missed a trick. Sticking it to the side of a tower PC or server box is even better. Of course, magnetising your case might not be a good idea, so if you do try it, we absolve ourselves of all responsibility for the integrity of your data.

There are a number of games available in this series, but the obvious king of the lot is chess.

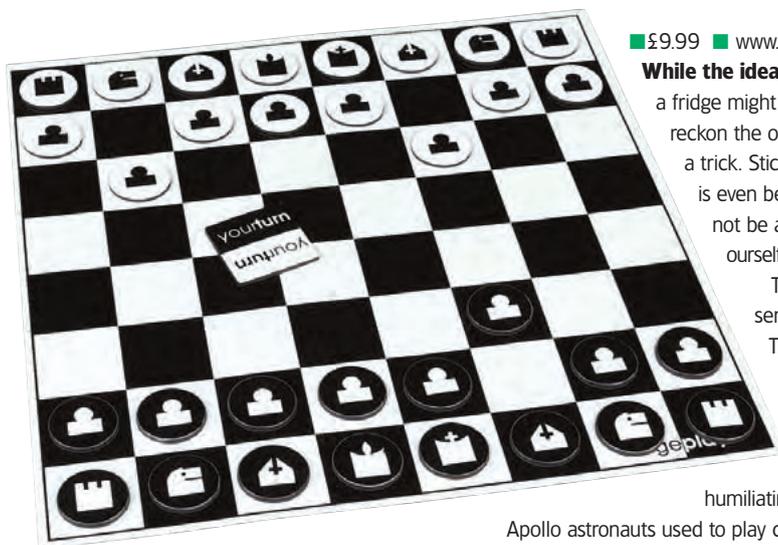
The board will stick firm to any metal surface, and the pieces stick limpet-fast to the board.

The design is minimalist, but the pieces are easy to recognise and play with. You'll have many a happy hour

humiliating your intellectual inferiors.

Apollo astronauts used to play chess with each other in their heads. If only they'd had a Fridgeplay board to slap onto the inside of the cockpit, there might have been fewer arguments.

❄️❄️❄️❄️❄️ 4/5



AD WALL

■ £12.95 ■ www.ad-wall.com

Why seek a high-tech solution

to your problems when you could make do with something a lot more tacky. The Ad-Wall is the amazing new solution to the age old problem of covering your monitor in so many post-it notes you can't see the screen anymore.

This monitor extension handily velcros to the side of your display, and the integrated 'Whiteboard' allows you to scribble illegible notes to yourself where they can be easily rubbed off and forgotten about later. If nothing else, it at least supplies another surface for you to apply post-it notes.



❄️❄️❄️❄️❄️ 2/5



HP 995C BLUETOOTH PRINTER

■ £285 ■ www.hp.co.uk

Printers, as a general rule, are not something to get too excited about. HP's 995c, however, is one of a kind – it's the only printer out there that has Bluetooth built-in. Rather than having your printer precariously balanced on a pile of books by your desk, you can now have it hidden away out of sight in a cupboard on the other side of the room. Not only that, but you can also throw away your parallel and USB cables – the only wire the 995c needs is one that goes into the mains. It's not only your computer that gets to play, either. With an increasing number of mobiles (and PDAs) coming with built-in Bluetooth and cameras (like the omnipresent Sony Ericsson T68i), you can take a snap with your phone and have it printed in seconds without getting your hands dirty. Not top-end in terms of print quality, but a fairly decent photo printer nonetheless.



SONY AIBO SPEED BOARD

■ £250 ■ www.eu.aibo.com

There's still only one AI robot you can buy that's any good, and that's Sony's four-legged AIBO. AIBO's always been cool – flashing lights at you when you say hello, stretching his legs when waking up and chasing pink balls like an international footballer – but the canine robot's one downside is that he's pretty slow. That's all changed, however, with the introduction of the hippest robot accessory money can buy: the AIBO Speed Board, a four-wheeled skateboard. In the interests of robot safety, you have to velcro AIBO on to the Speed Board, but after that he's off, using his paws to pick up speed and rolling around your house (offroad use, we're told, is not recommended). A special memory stick Speed Board program also lets it respond to skateboarding voice commands and change direction as you instruct. Absurd, but very, very fun.



ARCHOS JUKEBOX MULTIMEDIA 20

■ £350 ■ www.archos.com

Mp3 jukebox, video player, slideshow viewer – Archos' new player is one ingeniously-designed and incredibly multi-skilled little box. Outside, you get a 1.5 inch colour LCD for navigating your music and watching videos, inside you get a 20GB hard-drive for cramming your CD collection and video flicks onboard.

One of the other things that makes Archos' hybrid machine stand out from fellow jukeboxes like the iPod, however, is the inclusion of an expansion port.

Currently, you can buy a camera attachment and a USB 2.0 adaptor for this (otherwise you're stuck to slow old 1.1), or plug in one of the bundled CompactFlash and SmartMedia readers for transferring snaps from your digital camera. A TV-recording adaptor is planned for next February.

The Jukebox Multimedia 20 would be brilliant, but the tiny LCD isn't that great and the chunky, messy styling is just a tad embarrassing.



since 1997

**Developer Solutions
Provider**

Business Package

Ideal for businesses,
everything you need to start.

- * 200MB Fast UK Web Space,
- * Your Own Domain name, fees included
- * Supreme Control Panel
- * 30 Mail Boxes
- * Spam Relay Protection
- * Full Graphical Statistics
- * 3GB Data Transfer per month
- * SSH Telnet & FTP
- * RealMedia & WindowsMedia
- * Hosted in our UK Data Centre
- * Expert telephone & email technical support
- * Regular tape backups
- * 24/7 server monitoring
- * Same Day Setup

Powered by Zeus Web Server

Supreme Control Panel allows you to manage your web site and email using an easy to use web-based interface. You can manage your files, have complete control of your email, password protect directories, view your graphical statistics and more

Advanced features include:
CGI, Active Server Pages (ASP), PHP4

£169 per year

Dedicated Servers, co-location,
Reseller packages also available.

Hosting your web site with CWCS lets you benefit from our state of the art data centre, speedy network, expert technical support and friendly customer service.

**Our Services are Your
Business Solution**

tel 0870 703 1000

info@cwcs.co.uk

www.cwcs.co.uk



prices exclude VAT.



VOTE FOR THE BEST!

Nick Veitch introduces the awards where **YOU** choose the winners.

Forget 'reality', this is where your vote really does count. In the only awards of their type, it is in your power to reward those who have done the most for the Linux community. There are no secret panels of judges, no fixes, and as far as we are able to verify, no multiple voting shenanigans either.

That's what makes these awards so prestigious in the eyes of those who receive them – they are, from start to finish, decided by popular vote.

We have already run a nomination campaign on the website to determine the front runners in each category. Usually four or five candidates for each section have clearly had the

most nominations, and therefore they get onto the shortlist. In categories where the results are less clear, this year we have extended the shortlists to allow you to vote for the ones you really want, rather than exclude the close runners up. From last year's experience, it isn't always the case that the candidate with the most

nominations goes on to win the category after all.

This year you'll find there are a few new categories, and a few new people, organisations and projects to choose from. Enough of the chitchat though, it's time for you to consider your options. Without further ado, here are the categories...

Best Game

As with last year, this could have been any game for Linux, commercial or otherwise. The surprise here was that last year's winner, *Return to Castle Wolfenstein* was nominated so often we felt obliged to include it again. Although technically released in 2001, I suppose a lot of people may not actually have bought it until 2002.

Joining *RTCW* in this category is a game yet to be released! *Unreal Tournament 2003* has been doing the demo thing though, and is obviously being played by quite a few Linux gamers.

The bizarre but compulsive *Frozen Bubble* has put in a strong showing too. Were not sure what it is about different coloured bubbles that appeals so much to the community, but it certainly seems to be taking up a lot of your time.

Proving that you can't beat a classic, *Lbreakout2* gets the respect it deserves, and proving that Linux users are keen on alternative uses of technology, *Uplink* rounds off the nominations in this category.

- **RTCWolfenstein**
- **Unreal Tournament**
- **Frozen Bubble**
- **Lbreakout2**
- **Uplink**

Frozen Bubble



Unreal Tournament 2003



Hardware Support

Due to the diversity of nominations in this category, we had to extend the shortlist to an unprecedented eight contenders. Sun, for their Cobalt devices and the LX50, IBM for their work in general, supporting Linux on a variety of systems, and HP, mainly we suspect for their servers and printers, were some of the hardware vendors nominated here. Lexmark got a lot of mentions, mainly for printer support, and nVIDIA and Matrox were both cited for graphics drivers. Two big chip manufacturers also get in this category – AMD for their support of Linux on the Hammer, and Intel for Itanium, compilers and plenty more.

- **Sun,**
- **nVIDIA**
- **Matrox**
- **IBM**
- **HP**
- **Lexmark**
- **Intel**
- **AMD**

IBM s390



Sun LX50



Best development tool

As well as being able to use Linux, it's vital that we have good development tools – where will the software come from otherwise? There were some very interesting nominations here, but it's no surprise that the GCC compiler suite makes it here.

On the IDE side of things, *Anjuta*, *KDevelop* and *Kylix* all get mentions, the latter probably more for the new C++ functionality rather than anything else.

Emacs got too many nominations to ignore, and deservedly though unexpectedly making it to the list, *Autoconf*, without which managing the build process would be that much harder

- **Anjuta**
- **Autoconf**
- **Kylix**
- **Emacs**
- **GCC**
- **Kdevelop**

Best Advocacy

The best Linux advocacy award goes to the individual or organisation that has done the most to promote the Linux cause during the year. Usually these guys are selfless evangelists who travel the world or pop up on *Newsnight* at the drop of a hat to put forward the case for Free Software and the Linux way. It's no surprise that Luminaries such as Alan Cox the quixotic Stallman appear here. Bruce Perens managed to get himself in the news a few times too, first by threatening to violate the DMCA at a Linux convention, then by being made vice-president in charge of finding a new job when HP decided that, in fact, all publicity is not necessarily good publicity.

The only organisation on the list has done sterling work in the last year at promoting Linux, and Free Software. Alternately refusing to acknowledge its existence, then comparing it to communism, cancer, and alluding to it being unhealthy and "un-American" – these were just the opening shots in a concerted campaign to highlight the advantages of our favourite OS. The expert tactic of 'restructuring' their own licensing structure to make thousands pay more for MS products was the most often quoted reason for their nomination this time.

- **Microsoft**
- **Richard Stallman**
- **Bruce Perrens**
- **Alan Cox**

Richard Stallman





Best Server Software

In spite of a very active year in this sector, there is little change in the shortlist for best server software. The only real change here is that *CUPS* has been displaced by *Qmail* (I'm not sure what the developers have done, I still think *CUPS* is great!).

As well as *Qmail*, *Postfix* also got quite a few nominations and gives MTAs a 40% share of the shortlist.

Samba fans have registered their opinion on software which for many is absolutely vital to their mixed network environment – its importance should not be ignored.

The final two on the shortlist are probably more obvious elements of a standard server. *MySQL* continues

to be, for many, the database back end of choice, still going from strength to strength since it was open sourced a mere couple of years ago.

Apache may have figured in the 'best advocacy' category, as it has switched many a website from proprietary to free software. The latest release has not been without its problems though. Although a lot faster and more robust in many ways, the take-up of the latest version hasn't exactly been rapid.

- Apache2
- MySQL
- Samba
- Postfix
- Qmail

Best Security Software

Smoothwall



Your server is only as good as the security tools you use to keep crackers at bay. Security is always a popular topic in the magazine so it's no surprise we received lots of nominations in this category. What was a surprise was that there were some very clear leaders. Smoothwall walked this category last year, and thanks to new versions, appears here. But there is stiff competition in the form of *OpenSSH*, the secure shell server/client is very popular. Two intrusion detection schemes and the excellent *freeS/WAN* VPN system also feature.

- Smoothwall
- OpenSSH
- LIDS
- FreeS/WAN
- Snort

Desktop Application

This is a pretty open category, and there are many types of applications that could fit in here. It might be an MP3/Ogg player, it might be a graphics package, and it could even have been an obscure file manager. But it wasn't.

Xine, *XMMS* and *MPlayer* all represent the multimedia brigade. There have been many advances in all of these during the last 12 months, and it's certainly gratifying to see that Linux users are no longer left out when it comes to playing all

sorts of sound and video formats.

An interesting entry this time was *fluxbox*. Perhaps not the desktop application that everyone would think of, but rather a desktop in itself. Based on *blackbox*, this new windowmanager has some great features and is gaining popularity rapidly.

But could it possibly top *The GIMP* – one of Linux's real killer apps? Although work on the main code has been rather sporadic in 2002, there are plenty of interesting side projects going in, including *Gimp-film*.

- Xine
- The GIMP
- XMMS
- MPlayer
- Fluxbox

Business Software

In the best Business Software

category we are looking for the application that you'd be happy to bet your business on. It doesn't need to be flashy or be overloaded with features – it needs to do its job well and leave you to get on with yours.

Unsurprisingly, office suites feature highly here, including last year's winner, *OpenOffice.org* – which this year reached *version 1.0*, and continues to go from strength to strength – and *KOffice*, which has had a pretty quiet year, comparatively. GNOME was nominated for the office components and also its accessibility features (see *Linux Pro* this issue for more on that important topic).

Perhaps the most surprising entry was that of *CrossoverOffice*, which as you may know, essentially allows you to run Microsoft's *Office* suite on a Linux box. Is this a great step forward for Linux kind? Is it a great technology that ensures MS compatibility now, or a cop out? Your votes will decide!

- OpenOffice.org
- Evolution
- Crossover Office
- GNOME
- KOffice

GNOME



OpenOffice.org



Best ISP/host

Another tricky category to call. Many ISPs don't give any Linux support, some that do may not offer the best service. A variety of ISPs and hosting services were represented in your nominations so we have extended the shortlist to

encompass the most frequently mentioned. Since this is a new category there's no previous form to go on, so make your vote count!

- Telewest/Blueyonder
- Uklinux.net

- Speakeasy.net
- Pipex
- Demon
- Freeserve
- Positive Internet
- Rackspace

Best Internet Software

Mozilla



Once again this category is mostly a battle of Browsers. 2002 has definitely been a big year for *Mozilla*, in which we saw the release *1.0* version, and great strides being made in *1.1* and the *1.2beta* versions. But everyone else has been busy too. *Galeon* continues to develop apace, and has proven very popular among those who don't see why their browser should be as big as their kernel. *Opera* development on the Linux platform has kept pace with their other offerings and continues to be popular. New improvements like built-in Java are sure to increase its popularity further.

The default KDE browser, *Konqueror*, has also been updated more than a few times in the last 12 months. Its versatility as a file manager, browser, or both at the same time keeps it high on the list of apps people run.

But not all Internet software is about browsing. Instant messaging clients were also well represented by your nominations, but the clear leader was *GAIM*.

- GAIM
- Galeon
- Konqueror
- Mozilla
- Opera

Best Coffee

What, we asked you, would be your favourite caffeine-rich coffee beverage for those all-night coding sessions or mammoth network bugtesting episodes. Which cup of darkness keeps your eyes open and brain humming? Which purveyors of the wonderful bean should be rewarded for their prowess? There were, obviously, a huge range of potentials to draw on, but several lifted themselves from the pack.

The High St favourite Starbucks was the only retail chain to make it through into the final five, and was joined there by three big brands. Kenco proved to be the buzz for some, while CafeDirect's fair trading policies made an impression. Italy's favourite, Lavazza, may prove to be the Linux user's favourite too.

And just to be different, we received many, many nominations – too many to be ignored – for something that isn't coffee at all, but tea. PG Tips rounds off the five.

- Kenco
- Café Direct
- Starbucks
- Lavazza
- PG tips

Best Support Resource

Google



The clear leader by virtue of number of nominations in this category had to reluctantly be excluded. It's great that many of you wanted to nominate the *Linux Format* website, but it would be embarrassing to have to present the award to myself if we should win. I'm sure you can find a suitable alternative for your vote amongst this shortlist though, including as it does the venerable Linux Documentation Project, and the big three distro vendors, who all

received nominations for their own web-based support services.

A large number of people also wanted to nominate Google, whose Linux channel is ideal for tracking down that illusive documentation or newsgroup answer to a common or not so common problem.

- Google
- Linux Documentation Project
- Mandrake
- Red Hat
- SuSE

Enterprise Award

HP's RX series



We used to hear stories about Linux not being 'ready' for the Enterprise market. Not so much anymore, and the reason will, in no small part, be down to the winner of this category. Plenty of industry names to choose from, and all have been supporting and promoting Linux in the corporate space this year. IBM it seems, have their fingers in every pie going, and have continue to

promote services and Linux solutions on everything from their s390 down. HP often say that while the others talk about Linux, their the ones actually selling the hardware. Their new Itanium 2 powered servers might clinch it for them (or perhaps for Intel, who are also involved with not only promoting Linux, but using it themselves to design next gen chips).

Finally here, the newcomer is a collaborative effort on behalf of distro vendors SuSE, SCO and others to create a version of Linux that will be more attractive to corporate customers.

- IBM
- HP
- Intel
- Sun
- United Linux

Awards2002



NOW IT'S UP TO YOU!

And there you have the contenders for the LXF Awards 2002. They all certainly deserve to be on the shortlist, but of course, only one of them can win. Well, actually we had a tie in one category last year, but I can't see that happening again...

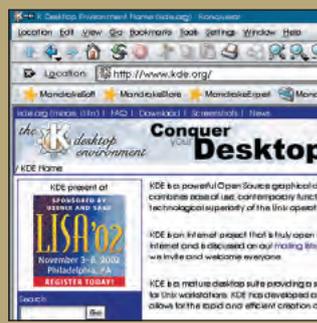
Anyway, the best part of it all is that your vote really does count – some of the incredibly close results last year just prove that. All it will take is a couple of minutes of your time to trundle over to:

www.linuxformat.co.uk/awards

and register your vote: you can only submit once, and as usual, we will be deploying as much technology as possible to ensure there are no multiple entries. So, what are you waiting for!

Free Software Project of the Year

KDE



Embedded Award

Trolltech's QTopia



While the embedded Linux

community might not be as visible as others, they carry out important work and, more importantly, enable the world to bring us cool devices. Trolltech for example, not only produce Qt, but also Qt embedded and the Qtopia development environment, helping people develop stuff like the Sharp Zaurus, which also gets a mention.

Montavista are a big name in embedded Linux, mainly for their distro, but also for work in all sorts of standards groups and organisations, helping to push the reliability of Linux in the embedded space even further.

Another distro that gets a mention here is uClinux, a reworking of the Linux kernel that removes dependency on an MMU – very useful for making use of all sorts of processors for embedded projects.

Finally for this category, the GNU Bayonne telephony project makes the shortlist. You may never have heard of it, but it is currently being tested in all sorts of phone network applications.

- Bayonne
- Montavista
- Sharp
- Trolltech
- uClinux

Best Distribution

SuSE



For some this is one of the most important categories. Certainly, the old 'which distro is best' argument fills up many bytes of data in our online forums. The question here should really be, which distro has improved the most in the last year?

SuSE and Red Hat have both upped the ante to 8.0 this year. While the SuSE leap was a more polished version of their previous offerings, Red Hat 8 seemed to go off in a totally new direction, with more focus on the desktop. But has the bold decisions taken to create an integrated desktop with the latest technologies paid off?

You can find out our opinion of Mandrake's latest revision elsewhere in the magazine.

And we mustn't forget Debian, which finally managed to release version 3 this year. Surely something that takes so long must be worth it in the end? We also see respect is due to the venerable Slackware and South American favourite, Libranet.

- Red Hat
- Mandrake
- SuSE
- Slackware
- Libranet
- Debian

As we said before, these are some of the most valued awards in the Linux community because they are all down to you. And they don't come any more prestigious than this. To carry off this category, the winner will have to have proved itself as quite simply a sublime piece of coding that the Linux world can't do without.

There are six nominees on the shortlist, all of which may be worthy of your adulation. First up, in alphabetical order, is Apache 2. The Apache group have indeed worked long hours bringing us this marvel. A new way of dealing with memory and processes promises even more efficient use of your server's CPU, and a new API for add-ons

promises to deliver even greater flexibility.

The GNOME project shipped version 2 this year and proved that, far from playing catch-up with KDE anymore, it has some great features of its own. KDE itself continued to grow, and the 3.0 release earlier this year offered some new features, mainly thanks to the transition to Qt3.

Mozilla, as we have already seen, has made a name for itself as a solid and reliable browser, and let's not forget that the journey to release has also spurred the development of some other great tools (e.g. Bugzilla).

OpenOffice.org has perhaps been even more successful than anyone could have imagined from the first

releases, culminating in sweeping the board when it came to our Ultimate Office software roundup.

Finally, and still to a certain extent controversially, we come to Wine – providing legacy compatibility for desktops that otherwise would not have switched to Windows. Thanks to the development of Wine and the code given back by the WineX project, it can now run countless Windows apps perfectly.

- Apache 2
- Gnome
- KDE
- Mozilla
- OpenOffice.org
- Wine

What on Earth is... USER-MODE LINUX?

David Coulson explains the User-mode Linux project and how Linux has been ported to Linux.

» Linux on Linux? Isn't that a little silly?

Well, yes, it is a little – Jeff Dike, the lead developer on *UML*, is a little odd, so there is likely method to his madness. There are lots of reasons why running Linux within Linux has many advantages over just running Linux on its own. Quite how one can benefit from User-mode Linux depends upon what one does with Linux. *UML* has a great many applications, which we will talk about later.

» So I run two Linux installations side by side at the same time?

No, one Linux kernel runs on another kernel. Only one of the Linux systems is running on the hardware, which is known as the host system. All of the other Linux systems which you run are *UML* instances, which are not actually aware of the hardware side of the machine and run purely within user-space, just as if they were a regular program, like *Mozilla* or *Vim*.

» Sounds familiar. Doesn't VMWare do this?

VMWare emulates a x86 system at the hardware level, allowing us to run Linux, Windows or many

other x86 operating systems on a machine. *UML* does not actually emulate hardware, but it instead provides a Linux kernel which runs purely in user-space. We can only run Linux with *UML*, so for people wanting to run Windows on Linux, *UML* is not an option at the moment. Of course, as *UML* is not running on emulated hardware, it runs much quicker than a Linux installation running on *VMWare*.

» Linux is running without hardware?

Yes, the *UML* kernel has no access to physical hardware on the system, unless the underlying Linux system which it is running on provides a way for it to access devices. Just like any other application running on a system, *UML* needs the host kernel to provide a method for it to communicate with devices connected to the system. As an example, for a hard disk, this would be the device node living within */dev*. Everything which is provided to programs running on the *UML* kernel is either virtual, or is provided by the host kernel.

» Let me get this straight. I can run lots of Linux installations, at the same time, on one box?

Indeed. With a single host machine running Linux, we can run as many *UML* instances as it can accommodate, depending on its hardware specification and the expected resource consumption of each of the *UMLs* we run. Of course, we can't use up more resources than actually exist on the host, so there is a limit.

» Sounds similar to IBM's s/390 mainframe, doesn't it?

Indeed it does. However, we all don't have a couple of million hidden under our bed or easy access to such powerful systems, so we have to make do with regular x86 kit. IBM's system uses a virtualisation system known as a *hypervisor*, which virtualises the environment for the kernel which runs on top of it. With the *hypervisor*, we can specify how much of the system's resources are allocated to each individual system which runs on it. Of course, due to the massive capacity of one of IBM's mainframes, we can run literally tens of thousands of Linux instances on the same box. Don't expect to do the same on a £300 Athlon system.

» So, how does UML do all this fancy stuff then?

The *UML* kernel runs just like a regular kernel, with 'physical' memory and a root file system to boot the system off. Of course, the *UML* does not use physical memory on the host, as it can't access the memory directly. There are two methods by which *UML* virtualises the system calls of processes running under *UML*, although at the time of writing, the new style of virtualisation is not currently available.

» Virtualisation? Sounds complicated.

Virtualisation is simply the method by which something creates a virtual environment for something else. At the moment, *UML* creates a process on the host for every process within the virtual Linux system and maps the the *UML* into the top of the process address space, just like the x86



WhatOnEarthUserModeLinux

kernel. Along with a process for each *UML* process, there is a single *UML* process called the 'tracing thread', which is sent the system call by the *UML* kernel running within the process space, and it proceeds to switch the state of the *UML* process so that the kernel executes the system call on the host and then the tracing thread switches the process back so that it returns from the system call instruction just as if it was running on a regular Linux kernel.

>>> Am I right in thinking that the UML process is actually running on the host system then?

Yes, it does. While *UML* processes can't make system calls to the host directly, the *UML* kernel will make system calls to the host on behalf of the *UML* processes, in the same way as a kernel running on hardware will translate the system call into a set of CPU instructions. *UML* processes actually execute on the CPU, so any binary we run on *UML* needs to be for the same architecture as the host.

>>> So UML running on a Pentium system has to run Pentium binaries?

Correct. We can't run a Sparc *UML* kernel and execute Sparc binaries if the host system is a Pentium or Athlon, just as we can't run Sparc binaries on a regular Pentium system running Linux. *UML* does not provide a method of running different architectures on a host, or otherwise creating alternative hardware.

>>> You said that UML can't access hardware. How on earth do I do anything with it then?

As the *UML* can't access any of our devices directly, we need to provide an alternative to each of the various hardware devices which Linux would normally use. The most obvious of these is memory, which is easily taken care of. Rather than using regular memory, *UML* creates a virtual memory file in `/tmp`, which is used as the 'physical' memory on the system. As with a regular machine, we can have swap storage in addition to our physical RAM.

>>> Surely if it uses /tmp for memory, it will be as slow as swap, since it's on disk?

If your `/tmp` file system is on disk, then *UML* will indeed run really slowly, since each time memory is accessed, it will be pulled from disk, rather than the much quicker RAM. However, Linux provides an alternative to this in the form of *tmpfs*, which is a little like a RAM disc, except that it will only consume the memory it needs. If we want a *UML* with 64MB of 'physical' memory, but we only actually use 8MB, then *tmpfs* will only allocate itself 8MB of memory on our system



>>> If I have 256MB on my host, I can run a UML with 512MB of memory?

tmpfs is allocated just like any other memory, so can be swapped out by the kernel VM, assuming enough swap space is available. One can run a *UML* with as much memory as you want, within the bounds of the Linux kernel, as long as sufficient storage space is available on the *tmpfs* file system and that the sum of your physical host RAM and swap is greater than this.

>>> How do I mount /tmp with tmpfs?

All it takes is a change to `/etc/fstab` to create a device using *tmpfs*. *tmpfs* is available in all 2.4 kernels and used to be known as *shmfs*. An example `fstab` entry would look something like;

```
none /tmp tmpfs defaults 0 0
```

As default, *tmpfs* will have a maximum size of half your physical memory, so if the host has 128MB, then your *tmpfs* have a maximum size of 64MB. We can supply an optional 'size' parameter, which sets *tmpfs*'s maximum size

```
none /tmp tmpfs size=512M
0 0
```

Even while `/tmp` is mounted *tmpfs*, we can change the maximum amount of memory which it can consume by remounting it.

```
mount -o remount,size=1G /tmp
```

>>> So that's memory sorted. What about storage?

The *UML* kernel has a module known as the *UML* block drive, which allows us to map a file system from the host onto a `ubd` device, then we can mount it within *UML*. Our file system can exist either on a partition on the host, on a LVM logical volume, or more usually, as an image file existing on a mounted file system on the host.

>>> My root file system for UML is a file on my real machine?

Yep. We just need a really big file, which has a file system on it and we can use it with *UML*. There

are many pre-built file system images available from the User-mode Linux web site and other locations, so it doesn't take too much to get a *UML* system up and running.

>>> If I want a 2GB root file system, I have to reserve 2GB of my host?

Not at all. We can create a sparse file, which is a file that has a specific size, but does not actually consume disc blocks until we write to it. If we create a 2GB sparse file and put a `ext2` file system on it, it might only use a few MB, if that, of actual storage space. This way, we don't end up wasting disc space, having it store lots of zeros in our empty file system.

We can create a 2GB sparse file and write a `ext2` file system to it with:

```
$ dd if=/dev/zero of=rootfs bs=1k count=1
seek=${2*1024*1024}
$ /sbin/mke2fs -f rootfs
```

>>> What about consoles and terminals?

Each virtual console in Linux is available with a **Ctrl-Alt-Fx** combination and has a *getty* running on it, so we can login and do things. As *UML* does not actually have anything for a virtual console to connect to, such as a VGA display, we need something else. The simplest thing for a console to connect to is an existing terminal and as default, *UML* will use the current terminal's `stdin` and `stdout` for `tty0`. For all other `ttys`, it will launch an *xterm* and the output from the terminal will appear within the *xterm*.

>>> That's pretty neat, but what if I don't have X?

As well as an *xterm*, we can point our consoles to a `pty`, which we can connect to with *screen*, or to a TCP port which we can access with something like *telnet* or *netcat*. Each console can be configured individually, so we can point some to `ptys`, some to *xterms* and some to TCP ports.

>>> Networking would be useful too. How do I do that?

UML supports no less than six different methods to provide networking to your virtual Linux system. Each of these have different capabilities and are more appropriate for specific applications, as well as requiring different capabilities on the host.

The most used network transport is through the `TUN/TAP` interface on the host, which is more commonly used by IP tunnelling programs such as *Vtun* or *OpenVPN*. With a tap device, we can create a virtual ethernet device within *UML*, and all traffic we sent to this will appear on the tap device on the host.

>>> If the traffic comes out of the tap device, what do I do with it then?

Generally, the tap device has an IP and the packets

are routed through our host onto our network, just like any other point-to-point connection. However, this means that our tap device needs an IP, as well as our *UML*'s device, so we end up wasting an awful lot of IPs if we have lots of *UML*s running on our system.

Instead, we can create an Ethernet bridge on the host which functions just like a virtual Ethernet switch. We can put lots of tap devices on this bridge, along with real Ethernet devices on the host, so we can set our *UML*s up with a direct connection to our network, as if they were a real system hooked up to our physical network switch.

» Okay, that's all the theory. How do I setup a UML system?

UML is actually purely a kernel patch, so all you have to do is grab the kernel tree, apply the *UML* patch, and off you go. Just like any other kernel, it needs to be configured with specific options and then built.

Once we've got our kernel from www.uk.kernel.org/pub/linux/kernel/v2.4/linux-2.4.19.tar.bz2, we can untar it with:

```
$ tar xjf linux-2.4.19.tar.bz2
```

The *UML* patches can be found at <http://user-mode-linux.sf.net/dl-sf.html> and at the time of writing, the latest patch is 2.4.19-18um, meaning the eighteenth *UML* patch for 2.4.19. With this downloaded, we can apply it to our kernel tree with *patch*:

```
$ cd linux-2.4.19
```

```
$ bzcat ../uml-patch-2.4.19-18.bz2 | patch -p1
```

Note that *UML* should not be compiled in `/usr/src/linux` and the *UML* kernel can be configured and compiled as a non-root user. We can now deal with our *UML* kernel tree just like any other, although to tell it to use the `um` architecture rather than `i386`, we add `ARCH=um` to the end of each command:

```
$ make mrproper ARCH=um
```

```
$ make menuconfig ARCH=um
```

```
$ make linux ARCH=um
```

Once `make linux` has finished, you'll end up with a 'linux' binary in your `linux-2.4.19` directory. This can be run just like everything else with:

```
$ ./linux
```

```
Linux version 2.4.19-18um
```

```
(david@maeve.dmz.davidcoulson.net)
```

```
(gcc version 3.2.1 20020924
```

```
(Debian prerelease))
```

```
#4 SMP Sun Oct 23 23:19:09 BST 2002
```

```
On node 0 totalpages: 8192
```

```
zone(0): 8192 pages.
```

```
zone(1): 0 pages.
```

```
zone(2): 0 pages.
```

```
Kernel command line: root=/dev/ubd0
```

```
Calibrating delay loop... 923.01 BogoMIPS
```

```
...
```

```
VFS: Cannot open root device "ubd0" or 62:00
```

```
Please append a correct "root=" boot option
```

```
Kernel panic: VFS: Unable to mount root fs on
62:00
```

```
<6>Stopping all CPUs...done
```

» How do I tell it where the root file system is?

We can pass a `ubdx` option, where `x` is a number between `0` and `7`, corresponding to a `ubd` device.

```
$ ./linux ubd0=rootfs ubd 1=swapfs
```

» What about consoles and networking?

Consoles are configured with the `conx` switch and `ethx` is used for networking. If we want all our consoles to be `xterms`, we can simply use `con=xterm`. An example command line might look something like:

```
$ ./linux ubd0=rootfs eth0=tuntap,tap0
con0=fd:0,fd:1 con=null
```

`tap0` can be created using the `tunctl` program from the `uml_utilities` package which is distributed on the *UML* site. To create the *UML*, we need to be root, plus the user which runs the linux binary needs to be able to read and write to `/dev/net/tun`, so `chmoding` that to `0666` is the simplest way.

Once *UML* is up and running, we can configure `eth0` with the appropriate IP address and setup routing, then *UML* will have complete network access.

» I don't have root on this system. How can I get networking?

The only transport which does not require root is `slirp`, which uses the `slirp` program to provide network connectivity. To use `slirp`, we just set our `eth0` device to use the correct transport and then configure `eth0` with the IP `10.2.0.15`, as per the `slirp` man pages.

```
$ ./linux eth0=slirp,./usr/bin/slirp
```

Not all network operations will function with `slirp`, as things like `ping` require root to open a raw socket, plus `slirp` is an IP only network device. However, for the most part, almost everything will work happily over this transport and as we don't need root to set it up, it's quite useful.

» UML sounds really cool. What are people using it for?

One of the main uses for *UML* is for kernel development, and some major kernel developers, including Alan Cox, use *UML* for testing their kernels. *UML* is a great development platform for kernel drivers which don't require hardware, as if the kernel panics, then we don't bring down the whole machine. Also, we can attach `gdb` to our *UML*, and just like any other

process, we can debug it while it is running.

Along with development, *UML* has been used for training and education, plus it has found a particularly useful application as a virtual hosting platform.

» How is UML being used for virtual hosting?

Rather than either having a dedicated server, which is a whole Linux system, or providing shell accounts, with *UML* one can provide each user with an entire Linux system, which is running on shared hardware. As one would expect, this is particularly flexible, plus it means that hardware costs are reduced, since we can now buy a system with hundreds of GB of disk space, rather than just 20GB.

There are currently two major *UML* hosting projects. One is by Bill Stearns, who has created `umlcoop.org`, as an alternative to a shared shell server for friends. Bill runs no less than twenty-five *UML*s on a dual-Athlon system with 2GB RAM and more disk space than you could shake a stick at and it has been proving particularly effective. More information on Bill's project can be found at www.stearns.org/slartibartfast

The other project is by myself, which provides *UML* hosting as both a virtual hosting platform for web and mail services, but also as a virtual server system. I've written a number of utilities to aid in providing a scalable *UML* platform, plus have written a comprehensive paper covering many aspects of *UML* hosting, which is at <http://uml.openconsultancy.com/>

» Where can I find more information on UML?

The official site for the *UML* kernel patch is at <http://user-mode-linux.sf.net/>. This is updated with all the latest patches for *UML*, plus is the main download location for all *UML* sources. This Sourceforge site also provides access to the mailing lists and documentation for the *UML* kernel.

There is also a community site, over at www.usermodelinux.org, which has all the latest news, tips and patches for *UML*, along with links and documentation. Just to show that *UML* is up to scratch, `usermodelinux.org` is hosted on a *UML* system and survived a slashdotting when *UML* was merged into 2.5.34. [LXF](#)



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

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.
```

THIS MONTH...

System programming

We continue on Linux system call interfaces with two classic mechanisms – pipes and signals **p66**

Perl

Improvements to Perl's pattern matching, from Perl 6 **p70**

Java

The JavaBeans framework gives us reusable components for manipulation by visual tools **p72**

OpenOffice.org

Mastery of all things spreadsheet is our objective for this issue **p74**

PHP >>

When to use templates and when not to use them. Plus why you may be better off 'rolling your own' **p78**

Process accounting

All things may be counted, if you can spare the resources to monitor your resource use. Find out when to do it, and how **p82**

Compiling

Understanding how software is compiled could help relieve some of those pesky installation problems **p86**



TIP OF THE MONTH!

If, like most of us, you spend any amount of time in a shell, you will have come to take the prompt for granted. By default it gives you useful information about the directory you are in, but there's so much more it could be doing.

To change your prompt, all you have to do is set the **PS1** environmental variable with a code. like this:

```
export PS1="\u>"
```

This will change your prompt to just your username followed by >. The **\u** is a special code that allows dynamic text to be added to the prompt – there's a list of all the special codes at

Changing prompts

the end here. By using special escape codes, you can also change the colour of the prompt, and the colour of the background:

```
export PS1="\e[32:1m\e[0m>"
```

For example, will change the colour of the previous example to a bright green. The **\e[0m** sequence returns the colour to normal, or the whole console would be green! Experiment with these, and you might find you come up with a more useful prompt than you currently have!

A quick list of some of the useful codes follows, but you'll find more details in the *Bash* man pages:

- \a** The ASCII bell character (x07)
- \d** Date in day-month-date format
- \e** The ASCII escape character
- \h** First level of domain name
- \H** Full domain name
- \l** Name of this terminal device
- \n** ASCII newline character
- \s** Name of current shell type
- \t** Time in 24-hour clock
- \T** Time in 12-hour clock
- \@** Time with am/pm postfix
- \u** Current username
- \w** Full directory path
- \W** current directory name
- \[** begin non-printing sequence
- \]** end non-printing sequence

PROCESS COMMUNICATIONS

Pipes and signals

Chris Brown continues his treatise on the Linux system call interface with discussion of two classic mechanisms in Linux – the pipe, and those mysterious things called signals.

Most sentient beings with an acquaintance with command line Linux know more or less what a pipe is, and how to make the shell create one (see panel below). For example, they know that the command

```
du /home | sort -nr
```

runs the programs *du* and *sort* concurrently, and connects the standard output of *du* into the standard input of *sort*. They possibly even call the vertical bar a “pipe symbol” because that’s the only thing they use it for. What most of them don’t know is what the shell has to do behind the scenes to make this happen.

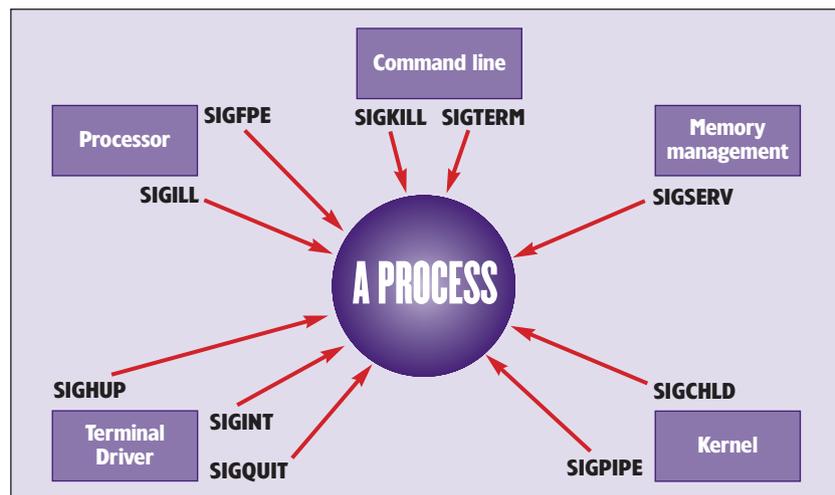
A pipe is a mechanism for sending a stream of data from one process to another. Pipes are uni-directional. Each has an upstream end, which you can write to, and a downstream end which you can read from. Each end is referenced by a file descriptor. You create a pipe with code like this:

```
int p[2];
pipe(p);
```

The `pipe()` system call creates the pipe and returns the file descriptors for the two ends in the two elements of the array `p`. The descriptor in `p[0]` refers to the downstream end, and `p[1]` refers to the upstream end. These are normal file descriptors and you can read and write them just as you would any other. Here’s the simplest program I can think of which illustrates this:

```
1 #define BUFSIZE 100
2
3 int main()
4 {
5     int count, p[2];
6     char buffer[BUFSIZE];
7
8     pipe(p);
9
```

Fig 1. Where do signals come from?



```
10 write(p[1], "through a pipe\n", 15);
11 count = read(p[0], buffer, BUFSIZE);
12 write(1, buffer, count);
13 }
```

The pipe is created at line 8. Line 10 writes a short message into the pipe, and line 11 reads it back into a buffer, which is then copied (line 12) to standard output. Having a process talk to itself through a pipe like this is pretty pointless – except to show the mechanics.

Pipes between processes

Setting up a pipeline between two processes is a bit more complicated. The key stages are:

- Stage 1: The original process A (the shell for example) creates a pipe, receiving descriptors on each end.
- Stage 2: Process A forks twice, creating child processes B and C. The children inherit the parent’s open descriptors, including those on the two ends of the pipe, so at this stage, all three processes have descriptors on both ends of the pipe.
- Stage 3: Each process closes the descriptors it doesn’t need. B closes the downstream end, C closes the upstream end, and A closes both. Note that B and C inherit their own copies of A’s descriptors, so that closing a descriptor in the parent doesn’t affect the child’s descriptors.
- Stage 4: Processes B and C `exec()` the programs they need to run. Again, the descriptors remain intact across the `exec()` call. B can then `write()` to the upstream end of the pipe, and C can read from the downstream end. Hey presto – interprocess communication!

But there’s a problem. Imagine that we’re the shell, trying to set up our original *du*-piped-into-*sort* example. The program *du* will write to its standard output. It doesn’t know anything about the pipe or the descriptor on the upstream end of the pipe that we want it to write to. Similarly, *sort* just reads from standard input; it doesn’t know about the pipe. So the shell has to fiddle around with the plumbing a bit. Specifically, it needs to arrange that in process B (which runs *du*), the standard output is connected to the upstream end of the pipe, and in process C (which runs *sort*), the standard input is connected to the downstream end.

For this we need the Linux programmer’s plumbing tool, a system call called `dup2()`. This call takes two descriptors as arguments, and makes the second descriptor refer to whatever the first descriptor currently refers to. For example, the code fragment:

```
int p[2];
pipe(p);
dup2(p[0], 0);
```

creates a pipe and makes the standard input of the process (descriptor `0`) refer to `p[0]`, the downstream end of the pipe.

[1] Except, that is, for the inhabitants of Carinoban 4, a planet in a more relaxed spiral arm of our galaxy, who instead invented the spray, an inter-process communication mechanism which causes the output from one process to be randomly distributed across all the other processes running in the computer. Any attempt to organise data actually results in it becoming more disorganised, with the result that the entire surface of the planet is now randomly covered with a layer of 0’s and 1’s.

Now (assuming you can remember back to the **fork()** and **exec()** calls we met in part 1 of this series) we know enough to do the complete job of piping *du* into *sort*. Here's the code:

```

1 main()
2 {
3     int p[2];
4
5     /* Create the pipe */
6     pipe(p);
7
8     /* Create the downstream process and connect
9     its standard input to the pipe */
10
11    if (fork() == 0) {
12        dup2(p[0], 0);
13        close(p[1]);
14        execlp("sort", "sort", "-nr", 0);
15    }
16
17    /* Create the upstream process and connect
18    its standard output to the pipe */
19
20    if (fork() == 0) {
21        dup2(p[1], 1);
22        close(p[0]);
23        execlp("du", "du", "/home", 0);
24    }
25
26    /* The parent closes both ends of the pipe
27    and waits for both children to finish */
28
29    close(p[0]);
30    close(p[1]);
31    wait(0);
32    wait(0);
33 }

```

The pipe is created at line 6. Line 11 creates the downstream process. Line 12 connects the standard input of this process to the downstream end of the pipe, **p[0]**, and line 14 has this process execute *sort*. Similarly, lines 20–23 create the upstream process, connect its standard output to the pipe, and run *du*. Lines 29–32 are executed by the parent process. It closes both ends of the pipe (remember, this does not affect the descriptors in the child processes) and waits for both children to terminate.

There is a certain elegance to this code which I find fascinating. The brevity and simplicity of the calls makes a telling contrast to the complexity of modern-day graphical applications, for example.

Pipes have a finite capacity and impose a loose synchronisation between the upstream and downstream processes. If the upstream process tries to write to the pipe when it's full, the process will block (*i.e.*, it will be suspended from running) until the downstream process has consumed some of the data. Similarly, the downstream process will block if tries to read from the pipe when it's empty.

Named pipes

The "anonymous" pipes we've just examined have a significant limitation – you can't use them to transmit data between two unrelated processes. The processes must have a common ancestor, which created the pipe and passed the descriptors down to its children.

You can also create named pipes, also known as fifos, which

```

chris@dopey:~ - Shell - Konsole
Session Edit View Settings Help
$ echo original message > sample.conf
$ ./sample
original message
original message
new message
new message
$

chris@dopey:~ - Shell - Konsole <2>
Session Edit View Settings Help
$ echo new message > sample.conf
$ ps ax | grep sample
1859 pts/1  $    0:00 ./sample
$ kill -HUP 1859
$

```

have a permanent entry in the filesystem, much like a file does. Any two processes can establish communication using named pipes, (provided, of course, they agree up front on the name of the pipe).

From the system call point of view, there isn't much to learn about using named pipes. You just open, read and write them like you would files. So we will demonstrate them from the command line. To follow along, you'll need to open two terminal windows on your desktop. In the first window enter the commands:

```

mkfifo foo
ls -l foo
cat foo

```

Notice that the **ls -l** command shows *foo* as being of type **p**, indicating that this entry in the filesystem is a named pipe. The *cat* command will open the named pipe for reading, and then wait for some other process to open the pipe for writing.

In the second terminal window, change to the same directory, and enter a command such as:

```
echo hello > foo
```

In the first terminal window you should now see the string **hello** being read by *cat* from the named pipe.

It's also interesting to try the test in the other order; that is, to run the *echo* command before *cat*. You will notice that the *echo* command hangs as it tries to open the named pipe, until some other process also opens it for reading. Thus, the named pipe forces what's technically called a rendezvous between the two processes – the first process to get there has to wait until the other one arrives.

Notice that pipes do not work across the network. Both processes have to be running on the same machine. (This is very different from named pipes in Windows, which have a name, a bit like a share point, which is visible within the network neighbourhood.)

Signals

Next, we'll turn our attention to signals. Unlike pipes, signals do not provide inter-process communication. Their basic function is to "get the attention" of a process when something unusual happens.

A signal is an event, sometimes described as a software interrupt, which is delivered by the Linux kernel to a process. Linux recognises almost 40 different signal types, each of which is identified by a symbolic name, such as **SIGINT**, and a number. You can get a full list of signals with the command

```
$ man 7 signal
```

A few of the more commonly encountered signal types are listed in the table overleaf.

Signals come from many places, as shown in **fig 1**. Some

Fig 2. The dynamic reconfiguration from the 'sample' program.



TutorialSystemProgramming

◀◀ signals originate from outside a process. For example, when a process is running in a terminal and the user enters **^C (Ctrl-C)** in the terminal window, the terminal driver responds by delivering a **SIGINT** to the process. Such signals are said to be asynchronous; that is, they are not directly related to anything your program is doing, and can occur at any point in the execution of the program. Other signals originate inside the process itself. A good example would be **SIGSEGV**, which is generated if an illegal

Name	Default Number	Action	Description
SIGHUP	1	T	Hangup
SIGINT	2	T	Interrupt from keyboard (^C)
SIGQUIT	3	T+D	Quit from keyboard (^)
SIGILL	4	T+D	Illegal instruction
SIGFPE	8	T+D	Floating point exception
SIGKILL	9	T	Kill signal, cannot be caught
SIGSEGV	11	T+D	Invalid memory reference
SIGPIPE	13	T	Write to pipe with no readers
SIGTERM	15	T	Graceful termination signal
SIGCHLD	17	I	Child process terminated

memory reference (known as a segmentation violation) is attempted. These signals are said to be synchronous; they are generated at a specific point in your program in response to something (usually something bad) that the program attempted.

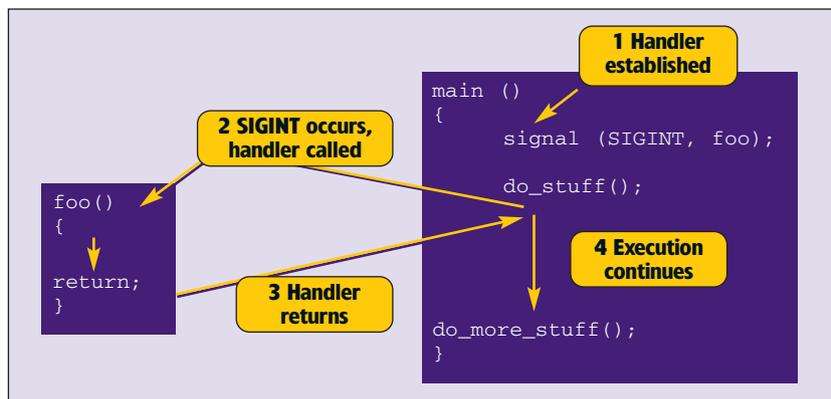
Default action for signals

So, what happens when a signal is delivered to a process? Well, it depends. First off, each signal has a default action which is what will happen if you don't specify otherwise. In most cases, the default action is to terminate the process. In the table these signals are marked with a 'T' in the Default Action column. Some of the signals (those that indicate that the program did something bad, and marked '+D' in the table) also trigger the generation of a file quaintly known as a core dump, which contains a memory image of the process at the time the signal was raised, and can be used for post-mortem debugging. A few signals, such as **SIGCHLD**, are ignored by default.

To demonstrate the effect of a signal terminating a process, try the following little program. (Line 7 is commented out, we'll add it back later.)

```
1 #include <signal.h>
2
3 int main()
4 {
5     int count;
6
```

Fig 3. Flow control on receipt of a signal.



```
7 /* signal(SIGINT, SIG_IGN); */
8 for (count=0; count < 10000; count++) {
9     printf("Working %4d\n", count);
10    sleep(1);
11 }
12 }
```

This program prints out one line a second, for 10000 seconds. Try running this program, allowing it to print out a few lines, then entering **^C**. A **SIGINT** will be delivered to the process which will immediately terminate, returning you to the command prompt.

Ignoring signals

For most signals, you do not have to accept the default action. You can specify what you want to happen when a signal of a specific type is delivered. Technically, this is known as "setting the disposition of the signal". The simplest way of doing this is with the system call **signal()**. One of the commonest uses of this call is simply to "turn the signal off"; that is, to tell the system to ignore it. For example, the call

```
signal(SIGINT, SIG_IGN);
```

tells Linux to ignore any **SIGINT** signals delivered to this process. Try adding this call in at line 7 of our last example, and re-run the program. (Don't forget to recompile it first!). You should now find that **^C** has no effect – the program will just continue. Don't panic ... you can still easily terminate the program by typing **^\
(control-backslash)** which will generate a **SIGQUIT** signal.

(Experiment for the brave. Try adding another call to **signal()** to make the program ignore **SIGQUIT** signals too. Restart it. Now neither **^C** nor **^\
I strongly recommend that you follow along and try these examples out. They won't take long to type in, and you'll get a much better feel for how signals behave.**

Establishing signal handlers

You can also nominate a function, known as a signal handler, which you want to be executed whenever a signal is delivered.

Let's extend our previous example to do this:

```
1 #include <signal.h>
2
3 void catcher(int sigtype)
4 {
5     printf("Signal caught\n");
6 }
7
8 int main()
9 {
10    int count;
11
12    signal(SIGINT, catcher);
13    for (count=0; count < 10000; count++) {
14        printf("Working %4d\n", count);
15        sleep(1);
16    }
17 }
```

The signal handler (a function we've called **catcher**) is defined in lines 3-6. Line 7 establishes **catcher** as the signal handler for **SIGINT** signals. If you build and run this version, and type a couple of **^C** characters at it, (followed by a **^\
like this:**

```
./signal3
Working 0
```

Working 1

Signal caught

Working 2

Working 3

Signal caught

Working 4

Working 5

Quit

When the signal gets caught, the handler function is called, and execution of the main program then continues from where it left off. The flow of control is illustrated in Fig. 3.

Sending signals

Signals can be sent to other processes with the **kill()** system call. The name is a bit misleading but derives from the fact that the default action for many signals is to *kill* the process. For example,

```
kill(1844, SIGHUP);
```

sends a **SIGHUP** signal to process **1844**. From the command line, the equivalent command would be:

```
$ kill -HUP 1844
```

Notice that the signal type and process ID arguments are in the opposite order. Don't you just love Linux?

It's probably worth noting that you can only send signals to processes you own, unless you're root of course, in which case you can send them to any process. There is one signal type that a process can neither catch nor ignore. This is signal 9, known as **SIGKILL**. If a process receives a signal of this type, it's certain death!

Four useful things

OK, so much for the mechanics. What are signals actually good for? We'll end up this month by presenting four useful things to do with signals.

First, you can arrange to ignore them. Whilst this may not seem terribly useful, knowing how to ignore signals is vital for writing robust code. If you're writing database code, for example, which caches some of the data in memory, you do not want to be vulnerable to being terminated by any signal that happens to come along. You want to have explicit control of when and how the process may be terminated.

Second, you can use signals to arrange for 'graceful termination' of a process. Conventionally the **SIGTERM** signal is used for this purpose. It is the default signal type sent by the **kill** command. Programs which create temporary files, or programs which put the terminal driver into a strange mode, are good examples of programs which should have cleanup handlers installed to delete the files, restore the terminal driver, or whatever.

Third, you can use signals to cause a program to re-read configuration data on the fly. This is an interesting application for signals and we will explore it in more depth.

There are many programs in Linux which read config data from a file when they start up. Daemons like *syslogd* and *inetd* are classic examples. If you want to change the configuration, it is not especially convenient to have to stop and restart the program, especially if it is busy serving a bunch of clients at the time. By establishing a signal handler to re-read the configuration file, the program can be configured dynamically by sending it a signal. It is conventional to use the **SIGHUP** signal for this purpose.

[An aside: The "HUP" in **SIGHUP** is short for "hang up". In the early days of Unix, **SIGHUP** was the signal generated by the terminal driver on a dial-in line if the modem lost carrier; that is, if whoever had dialled in decided to "hang up" the phone. Its

purpose was to ensure that if a user disconnected without logging out, his shell would receive a **SIGHUP** and would terminate, ensuring that the next person to log in on that line would not find himself talking to the previous user's shell. Dial-up logins are rare these days, being largely replaced by ppp connections, and the **SIGHUP** signal has a new purpose as the "reconfigure" signal.]

We'll illustrate with a program called 'sample' that prints out a message every second. When it starts up, it reads the message from a file called *sample.conf*. If the user wants to change the message while the program is running, he edits the file then sends the program a **SIGHUP**. The program re-reads the file, thus updating the message. Here's the code.

```
1 #include <signal.h>
2 #include <fcntl.h>
3
4 int count;
5 char buffer[100];
6
7 void read_config_file(int sigtype)
8 {
9     int fd;
10
11     fd = open("sample.conf", O_RDONLY);
12     count = read(fd, buffer, 100);
13     close(fd);
14 }
15
16 int main()
17 {
18     read_config_file(0);
19     signal(SIGHUP, read_config_file);
20     while(1) {
21         write(1, buffer, count);
22         sleep(5);
23     }
24 }
```

Remember that execution of a program always begins with the **main()** function. At line 18 we call **read_config_file()** to establish an initial configuration, and at line 19 we arrange that this function will be called again each time we receive a **SIGHUP**. Lines 20-23 represent our main service loop; in this example we just print the text we read from the configuration file every five seconds. In a real-world example we would use the data we read from the file in some less trivial way.

The screenshot (**fig 2**) shows the dynamic re-configuration in action. We begin in the left window by writing a line of text to the configuration file *sample.conf*. Then we launch the program, which begins to print out copies of the message. Now, in the right window, we write a new message into *sample.conf*. This does not, by itself, change the output from the running program. We need to send it a **SIGHUP** to tell it to re-read the file. First, we must find the process ID of the the program, using *ps*. Knowing this, we can then send a signal to the process from the command line with the **kill** command. At that point, the output from the program changes to the new message. (It's hard to see the sequence of events from the screen shot. Try it out for yourself.)

Our fourth and final suggested use for signals is to force a printout of debugging or status info on demand. The idea is to have a way of querying the current state of a long-running program. The program keeps track of its status in a set of variables in memory, and a signal handler is written and installed to print them out. 

NEXT MONTH

In the final part of this series next month, we'll take a look at the system calls generally known as "sockets" which support the writing of client/server applications using TCP/IP, and we'll write our masterwork for the series – a simple web server.

REGULAR EXPRESSIONS

A match made in heaven



Charlie Stross goes dumpster-diving in Larry Wall's in-tray and comes back with an explanation of how Perl 6 is going to improve on Perl 5's pattern matching.

Perl wasn't the first programming language to make a big deal out of using patterns to manipulate strings, but Perl's approach is one of the most powerful, and goes a long way towards explaining the language's enduring popularity. Older Unix tools, starting with the *ed* text editor and the *Bourne* shell, used an abstract language called "regular expressions" to define the shape of character strings rather than their actual content. The *sed* streaming editor and later *awk* pattern-matching language generalised the concept further, and Perl started out with a superset of *awk*'s pattern-matching system.

Back in the days of *sed* and *awk*, most text that you came across on a Unix system consisted of plain old-fashioned unstructured ASCII, with line breaks. Perl's regular expression system was developed to cope with this sort of material. For example, it's dead easy to write a basic expression that matches an English sentence:

```
/ [A-Z] # first letter capitalised
[a-z] # second letter lowercased -- avoids catching initials
(.+)? # one or more subsequent characters, minimal match
[.\!\? ] # end of sentence
/xs
```

This uses Perl 5.6 vernacular – the **x** modifier after the end of the pattern indicates that whitespace and comments are allowed in the pattern, and the **s** modifier indicates that we want to do multiline matching, in which a newline is matched by the **.** character. It also uses minimal matching – **(.+)?** matches the shortest sequence of one or more characters preceding the following (end of sentence) pattern. Note that this approach isn't perfect – we might run into problems with quotation marks around periods – but it's a fairly robust expression for dealing with blocks of ordinary ASCII text. Where it breaks down is in two places – dealing with non-ASCII text, and dealing with structured text.

Non-ASCII text came up first, in the form of unicode and related multi-byte codesets. Some support for Unicode was bolted into Perl 5.6, in particular the ability to set the number of bits to match for a character and to use predefined character sets, but it fundamentally broke the clean simplicity of the earlier pattern matching model by essentially making it modal.

But the biggest challenge is structured text, such as HTML or XML (or any programming language). Structured text is *recursive* – i.e. it consists of elements which may contain other, nested, elements. An example is the common problem (for Perl regex wizards) of ripping the plain text contents out of an HTML file:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
```

```
<HTML>
<HEAD>
  <TITLE>Example file</TITLE>
</HEAD>
<BODY>
  <H1>A simple file</H1>
  <P>
    Hi there! I am a <B>very <U>simple</U> example of a
    <I>structured</I></B> file.
  </P>
</BODY>
</HTML>
```

The goal of extracting the textual contents of this file is to produce something like this:

```
Example file
Hi there! I am a very simple example of a
structured file.
```

A naive approach is to simply rip out anything like an HTML tag:

```
$file =~ s/<.+?>/g;
```

This works up to a point. But what if we want to extract only the text that is in italics and boldface, but not in italics or boldface? To do this, our pattern matching system needs to know something about the context in which it is trying to match a string – which tags it is enclosed by. This rapidly gets hairy, because the current string could be nested within any number of tags.

The Perl 5 way of dealing with this sort of problem is to use a parser module, such as **Parse::RecDescent** or **Parse::YAPP**. These parsers allow you to define the structure of a file from the bottom up (specifying what elements look like), and associate actions to take (in the form of code to execute) which each element. But this is rather clumsy, and goes against the Perl 6 goals of (a) simplifying things, and (b) turning Perl 6 into a language for writing application domain-specific mini-languages.

The Perl 6 way

The first thing to notice about Perl 6's approach to regular expressions is that the **x** modifier is turned on by default. Introduced in Perl 5.6, this modifier means that whitespace is allowed in regular expressions – if you want to match whitespace you have to use the **\w** (whitespace) pattern or backslash-escape it. By the same token, comments are allowed in regular expressions to make them more readable – which makes sense when you consider that the Perl 6 system is geared towards writing full-scale parsers.

Next, there's the addition of the regular expression constructor, **rx//**; this generates a new expression which can be stored in a scalar variable. For example:

```
my $pattern = rx / <$tag> # match whatever the rule $tag matches
               .*? # minimum-match of any character string
```

```
<Sendtag> # match whatever the rule Sendtag matches
/;
```

(Note that the **rx** constructor we use to build a regular expression object isn't a quoting mechanism (like the older **qq//** syntax); we can refer to **\$tag** and **Sendtag** before they're defined and they'll be interpolated whenever we actually apply **Spattern** to a target string at runtime.)

It's also worth noting that the modifiers that specify how a regular expression should be processed are all changing. You can stack a bundle of modifiers in front of a pattern delimiter, to do things like specify case-insensitivity, repeatedly match as many times as possible (changed to **e** from Perl 5's **g** modifier), pretend to be Perl 5 (for backward compatibility), and so on.

It's new terminology time in Perl land: we now have named patterns, called *rules*, and we can embed rules in each other using **<>** to enclose them. If we omitted the angle-brackets and simply put **\$tag** and **Sendtag** in our expression, they would be interpolated into **Spattern** when it was executed – but as strings, not as rules.

We can apply other regular expression modifiers to rules: for example, **<\$tag>{2,3}** matches when the pattern in **tag** either two or three times. Another major twist in Perl 6 is the ability to match arrays or hashes directly:

```
@colour = ('red', 'blue', 'green', 'orange');
$text =~ / @colour /;
```

Within the regular expression, **@colour** is replaced by a pattern matching any of its contents – in Perl 5 we'd have had to construct a regular expression from it first:

```
@colour = ('red', 'blue', 'green', 'orange');
$colour = join '|', map { quotemeta $_ } @colour;
$text =~ / (? : $colour ) /;
```

Now things begin to get peculiar. In Perl 5, square brackets enclosed character sets. In Perl 6 they don't; they enclose a *noncapturing group*. Character sets like **[A-Z]** may work in ASCII, but they're less useful in unicode where they'll stop working if you accidentally blunder into the wrong language find yourself processing Greek or Chinese text by mistake. A noncapturing group is one that matches some text but doesn't capture it (for interpolation). You can still do character classes in Perl 6, but you need to put them inside the metasyntactic marker (angle brackets). We do have some special names for frequently used characters, though: **<sp>** for space, **<ws>** for whitespace, **<lt>** and **<gt>** for angle brackets, **<dot>** for a period, and so on.

Using metasyntactic markers and rules lets up build arbitrarily complex trees of patterns – essential if we're going to write a recursive-descendant parser. But as anyone who's written a parser knows, matching patterns is only half the problem; if you're matching some input against a complex ruleset, and part of a ruleset has failed, you need to have a mechanism to control backtracking. Perl 5 controlled backtracking implicitly, but Perl 6 has a much more fine-grained mechanism.

We can control backtracking in Perl 6 by using the **:**, **::** and **:::** operators after a rule, but within a noncapturing group. For example:

```
[ <$rule1> : <$rule2> <$rule3> ]
```

This attempts first to match **\$rule1**, then **\$rule2** **\$rule3**. But the colon operator means "if the preceding match fails, don't bother backtracking to the previous element." A double-colon means "don't bother backtracking within the enclosing group", and a triple-colon means "if we have to backtrack here, the entire rule fails (including but not limited to the current group)". And there's a special directive named **<commit>** – if we have to

backtrack through a commit, the entire match fails immediately. (This is what we'd use in throwing a syntax error from inside the parser of a compiler, for example.)

More fun new features

In Perl 5, it was pretty damn hard to read from a file handle and apply some kind of pattern-match to the process as we did it. There were workarounds (such as **IO::Stringy**), but in general you couldn't apply pattern matching to file handles directly without using a scalar as a buffer. Perl 6 fixes this by letting you bind an input stream to a scalar and match against it:

```
my $text is from($^ARGS); # Bind scalar to input stream
if $text =~ /<$pattern>/ {
    # do something
}
```

In addition to letting us do fun things with input streams, Perl 6 makes it easier to manage complex pattern-matching tasks by adding some new declarations – grammar and rule. A grammar is the pattern-matching equivalent of a Perl module; a collection of rules grouped together in curly brackets and denoted by a name, in their own namespace. Rules within a grammar are named, and may be referred to as **grammarname.rulename**, just as methods within a module are referred to as

Modulename::methodname. The grammar declaration either applies to the block immediately following its name, or to the rest of the file. For example (from Damien Conway):

```
grammar HTML {
    rule file :iw { \Q[<HTML>] <head> <body> \Q[</HTML>] }
    rule head :iw { \Q[<HEAD>] <head_tag>+ \Q[<HEAD>] }
    # etc.
} # Explicit end of HTML grammar
```

We can match against named rules by putting the name in angle brackets, for example:

```
$file =~ /<HTML.bold_tag>/
```

The analogy between rules and subroutines goes even further; like subroutines, we can pass arguments to rules (including the new Python- style of argument passing with name/value tuples that Perl 6 is assimilating).

A rule-based parser in Perl 6 builds a hierarchical data structure of results as it goes along. Whereas in Perl 5 the result of a pattern match would be available as **\$1**, **\$2** ... **\$n** for substitution, and **\$0** as the entire string, in Perl 6 we have a hash, too. If we match a sub-rule, its contents become accessible via **\$0{rulename}**; this works recursively, so the result of running a bunch of recursive rules against a structured file is actually a parse tree of hashes.

At this point, we've only begun to scratch the surface of Perl 6's regular expression language. We can embed arbitrary code blocks in rules, so that if we're writing a web browser (for example), then encountering a tag name can trigger some graphical action (such as rendering its contents in boldface). The possibilities are huge – basically the parser modules (such as **Parse::RecDescent**) are now obsolete, with equivalent functionality built into the core language. Perl 5 needed no *lex*-workalike, but Perl 6 builds in the same functionality as *yacc* (albeit better designed). If you want to read more about writing compilers and parsers in Perl 6, Damien Conway's exegesis (see www.perl.com/pub/a/2002/08/22/exegesis5.html) is currently the place to start; in the meantime, rest assured that although your existing regular expressions will still work (with the **:perl5** modifier to tell Perl 6 to be backward compatible), things just got a whole lot more powerful. [LXF](#)

A BIT ABOUT BEANS

Speaking Java

What's so special about JavaBeans? Find out with this introduction from **Richard Drummond**.



Visual development environments – tools which let you build applications largely by point-and-click – are all the rage on many language platforms, and Java is no exception in this regard. The basis for Java's component architecture – the building blocks which can be manipulated and assembled with such visual tools – is the JavaBeans framework. But what is a Java bean? How do you make use of JavaBeans components? And how do you create your own? These are some of the questions we will begin to look at in this article.

The basics of beans

The JavaBeans technology provides a specification for creating reusable Java components that can be manipulated with visual tools. A Java bean is component which meets that spec. There's nothing mysterious about a Java bean – there's no special **Bean** class that must be inherited, no *Beanable* interface to inherit. A bean is simply a regular Java class that obeys certain conventions, mainly naming conventions, as documented by the JavaBeans specification.

If you are a Java programmer, you've already used beans, even if you don't realise it. Many of the important classes in the standard Java class library are beans, such as the *AWT* and *Swing* widgets. Two key points to notice here: you don't have to use a visual bean builder tool to use beans; and beans don't necessarily have to be visual components such as widgets. A component which encapsulates a database connection could equally well be a bean, although it doesn't have any directly visible aspect at run-time.

The design-time interface of a bean is equally as important as the run-time interface that a Java program that uses the bean sees. The tools to create and use these design-time interfaces

Bean resources

Where to go for information and downloads

<http://java.sun.com/products/javabeans>

The JavaBeans homepage at Sun. Contains links to documentation, tutorials, tools, and directories of bean vendors.

www.netbeans.org

NetBeans is the open-source project behind Sun's *Forte for Java* IDE.

www.borland.com/jbuilder

Jbuilder is Borland Java IDE. The Foundation edition is available as a free download.

are provided by the JavaBeans API, implemented in the package *java.beans*. This API is intended for those who design bean tools and bean vendors, it can be safely ignored by the programmer who just needs to use beans. As well as the bean itself, the design-time interface of a bean can be augmented with various auxiliary classes which can provide extra info to the bean builder tools.

A bean is typically packaged up with any of these auxiliary classes, docs, etc., and distributed as a jar file. A user can then import this jar file into a development tool, such as *NetBeans* or *JBuilder*, to manipulate it, use it, and incorporate it into their programs.

Bean features

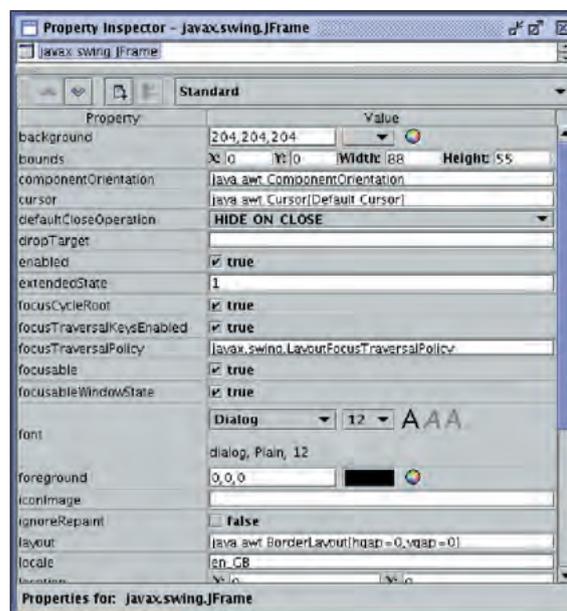
If a bean is simply an ordinary Java class, how does all this work? The following key features are the secret to the Java bean's usefulness.

INTROSPECTION

Tools which use the JavaBeans API to manipulate beans do so via introspection. Introspection, as provided by Java's reflection mechanism, is Java's ability to obtain get run-time information about classes, such as the names and parameters of methods. The **Introspector** class, part of the JavaBeans API, uses reflection and the naming conventions of the JavaBeans specification to analyse the interface that a bean provides and so present you with a GUI to manipulate it. As we said, as well as the bean itself, each bean can have several auxiliary classes which provide extra design-time information and tools. The **BeanInfo** interface can be implemented by such a class to provide programmer-friendly information about a bean, such as descriptions of its properties methods and events, and even an icon to represent the bean in a builder tool.

PROPERTIES

Properties are those elements of a bean's state that can be manipulated visually, for example, the height and width of a widget or the URL of a database connection. Properties are accessed by methods which obey a strict naming convention, so-called 'getter' and 'setter' methods. A builder tool discovers



A tool such as Sun's **BeanBuilder** provides you with GUIs to visually modify a bean's properties.

a bean's properties via introspection, and so allows a user to manipulate them.

CUSTOMISATION

The exposed properties of a Bean can be customised at design time. A bean tool will provide an appropriate interface to manipulate a property depending on its class. For simple classes of property standard controls such as a string and chooser widgets will suffice, but for more complex classes, bean vendors can supply custom GUI controls for setting a property's value. Thus, some kind of colour-palette tool could be provided for setting properties of class **Color**, for example. For complex beans, a **Customizer** class can be supplied to provide a complete GUI for manipulating the state of that bean. An example here would be some kind of wizard to lead you through the setup of a bean's state.

COMMUNICATION

Beans can communicate with one another by broadcasting and listening for events. The JavaBeans event model is based on the *AWT's* event mechanism, so should already be familiar. In this type of communication, one bean is a source of certain types of events. Other beans which are interested in particular events that this bean can generate register themselves with it as listeners, so are notified when that event occurs or is 'fired'.

PERSISTENCE

Beans can use Java's object serialization mechanism, as implemented by the **java.io.Serializable** interface, to save and restore a bean's customised properties. You can customise a bean in a builder tool, serialise it to a file, and then reload these changes to recreate the customised bean at a later date.

METHODS

As well as accessor methods for manipulating properties, beans have 'regular' methods for performing operations, just like any other Java class. There are no special conventions for naming of a bean's methods. All public methods will be visible to a bean builder tool.

Property conventions

To reiterate, a Java bean is a normal Java class which adheres to certain conventions, and it is these conventions that let a builder tool discover its interface. We don't have the room for the complete list here, but we'll cover the naming rules for properties. There are no special restrictions on the class name that a bean may be given.

When a bean defines a property named **P** (a property's name is the name that will be displayed in the bean builder tool) of type **T** (where **T** can be a scalar type, either a primitive type or a class) its getter method must have the following pattern

```
public getP( )
```

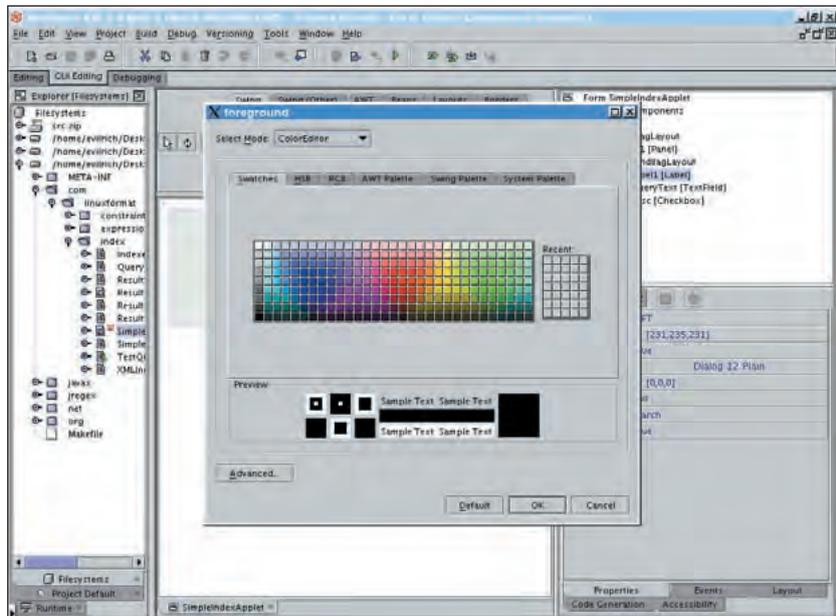
and its setter method must be of the form

```
public void setP( T )
```

Property accessors can throw any type of exception, checked or unchecked.

Note that you don't have to provide both setter and getter methods for a property. If a property is to be read-only, then a setter method shouldn't be provided. Also, these accessor methods should also obey the usual Java conventions about the naming of methods. For example, a property **width** of type **int** would be accessed via the methods:

```
public int getWidth( )
```



The **NetBeans** project is a full Java IDE with tools for visually assembling and manipulating beans.

```
public void setWidth( int )
```

Notice how we capitalise the first letter of the property name in these methods.

As a special case, for *boolean* properties, you can alternatively name its getter method with the pattern:

```
public boolean isP( )
```

For example, for a boolean property **displayable** this method would be:

```
public boolean isDisplayable( )
```

As well as scalar properties, a bean can have indexed properties or properties of array type. In this case, getter and setter methods can be provided to access the array as a whole or a single element. For a property **P** of type **T** its array accessors must meet the patterns:

```
public T[] getP( )
```

```
public void setP( T[] )
```

while its element accessor must conform to the patterns:

```
public T getP( int )
```

```
public void setP( int, T )
```

Bound and unbound

As well as simple and indexed properties as we have seen, a bean's properties can additionally be 'bound' or 'constrained'. A bound property is a property that supports the broadcast of property change events to registered listeners whenever the value of that property is modified. Constrained properties take this one step further and allow a listener to veto a change to prevent that property value being modified.

Bound or constrained properties must meet the same naming conventions as standard properties, but since this does not allow the introspector to tell the difference between a bound and unbound property, the **isBound()** method of the **PropertyDescriptor** object returned by the beans **BeanInfo** class should answer **true**.

A bean that has bound or constrained property must provide standard methods so that listeners can register and unregister for notification of property change events. We'll look at the JavaBeans event mechanism in more detail next time, but this is easy to implement because Java supplies all of the infrastructure for the event mechanism. [LXF](#)

NEXT MONTH

Following on from this brief introduction to JavaBeans concepts, next time we'll be doing some real work and implementing a **JavaBean** and its accompanying **BeanInfo** class, and looking at how a bean can be packaged for distribution. We'll also provide the finished code for the **MultiList** widget that we started writing last time.

SPREADSHEETS

OpenOffice.org for power users

This month **Neil Lucock** gets to put his shoes back on as he attempts to count past 20 with Open Office.org's versatile and powerful Calc spreadsheet.

We'll start off with a quick look around the selection of icons that *OpenOffice.org* (OOo) displays when we start a spreadsheet. OOo starts with the word processor, go to Tools>Options>Load/Save, then choose "General" if you want to alter this. You get the same basic interface for each part of the suite, it just displays a few different icons (and, in the case of the spreadsheet, the cells) depending on what type of document you make. OOo is easy to get used to, most of it remains the same no matter what you are doing. However, don't let the simplicity and price make you think OOo isn't up to the job. It is a professional product with a comprehensive range of tools. If you can use *Excel*, you can do the same job in OOo *Calc*. You'll just save money.

Toolbars

OOo gives you a couple of toolbars to access useful commands. Toolbar 1 has a currency format on the left.



Defining how *Calc* displays any money values. Format>Cells will get you to this dialogue. Click on the Alignment tab to alter the contents to an odd angle in the cell.

Spreadsheets allow you set defaults for how data entered into each cell is displayed. Normally the default is General. It displays numbers as they are. You can highlight a cell (or a Row, Column or group of cells) and change how the data is treated. Currency settings are changed under Format>Cells... by choosing Numbers from the tabbed box. OOo supports the Euro as well as the original European (and non-European) currencies. You can also define how you want your money to display, so if you want a couple of leading zeros before the decimal point if the amount is less than a pound, you configure it there.

You can also mandate a variety of options from the same dialogue box. OOo benefits from a high degree of integration of the different parts of the suite. You can easily make the contents of a cell display at 90 degrees to the rest of the sheet if you need to. Not a vital feature, but nice to have.

Useful

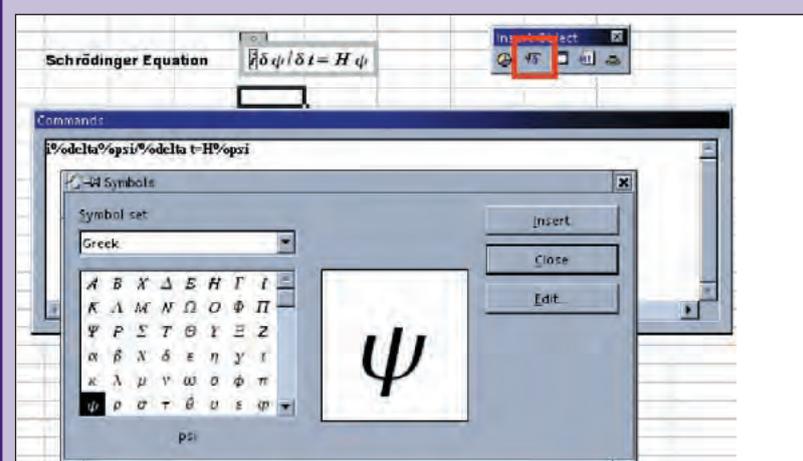
The next icon shows a percentage sign. If you type 1.23 into a cell and click the icon, you'll get 123% and typing 0.6 gives 60%.

The third icon, a dollar sign, a percentage sign and a backwards arrow underneath, removes the formatting the two previous icons imposed. It's a sort of specialised undo tool. The next two icons add and remove decimal points to the number entered in the cell.

The rest of Toolbar 1's icons are for cell formatting. You can move the contents to the left or right within a cell, put borders around all or just part of it and change the background colour. If you have a deep cell (perhaps because you have inserted a picture which has made all the cells on that Row wide enough to

Math

Displaying formulae

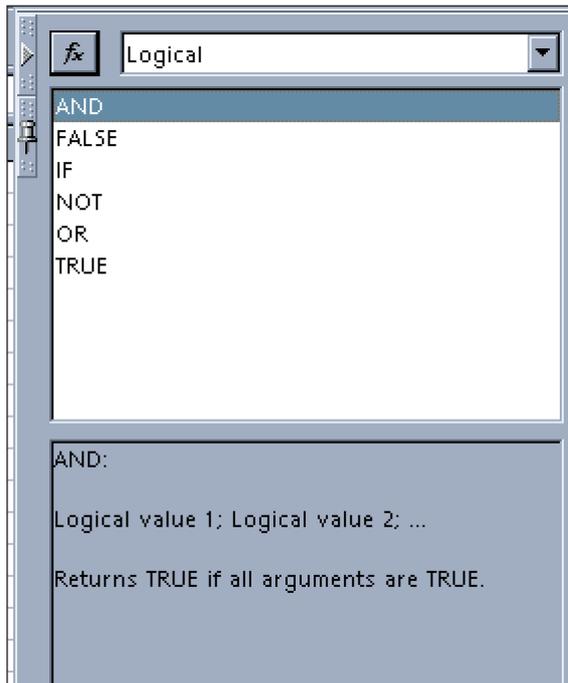


Math displays formulas, it does not work them out. That's your job.

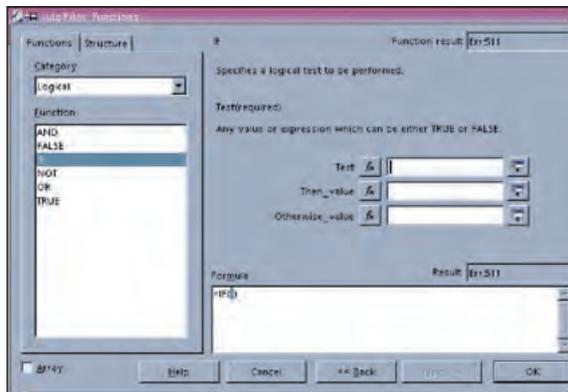
OOo includes *Math*, a program to insert formulas into your spreadsheet. Click on the icon with a ? on it. *Math* changes the toolbar and displays a box to allow you to enter your equation.

Note that this is not a utility to calculate the

formula shown, it is a way of displaying the formula, nothing more. The utility really is a specialist word processor applet, which extends the usefulness of OOo if you have to reproduce complex formulas for publication.



The Function quick reference gallery, usually hidden over on the far right of the spreadsheet. The types are usefully categorised and there's a reminder on how to use each one.



The Functions Autopilot (wizard), accessed through the Insert>Function menu.

fit it) you might want to alter the vertical display position of your figures. The last three icons on Toolbar 1 let you move the contents to the top, middle or bottom.

All Cells in a spreadsheet have a reference, with letters across the top and numbers down the sides. If you need to go beyond Z it uses AA, AB, then BA, BB etc. The Cell you are working on is displayed on the left of Toolbar 2.

The icon that looks like a keyboard launches the Function autopilot. This tool offers help in creating various formulas. You need to know what you are doing mathematically, it will not make you into Einstein, but if you need to work out inverse hyperbolic cotangents it will help you to get the data in the right order.

SUM thing

The **SUM** (?) character is next. It adds up the list of numbers above it, (by default) so make a few numbers in a column, click in the cell below them and it adds them all up and displays the result. If you want it to add up a Row, click in the Cell where you

The Help System

An unobtrusive friend

Date & Time Functions

These spreadsheet functions are used for inserting and editing dates and times. The functions are: **WORKDAY**, **YEARFRAC**, **DATE**, **DATEVALUE**, **EDATE**, **TODAY**, **YEAR**, **NOW**, **WEEKNUM**, **WEEKNUM_ADD**, **MINUTE**, **MONTH**, **EOMONTH**, **NETWORKDAYS**, **EASTER_SUNDAY**, **SECOND**, **HOUR**, **DAY**, **DAYS**, **DAYS360**, **WEEKDAY**, **TIME** and **TIMEVALUE**. They are described in this order below. Call the 2nd page of AutoPilot: Functions to compare the described function.



OpenOffice.org internally handles a date/time value as a numerical value. If you assign the numbering format "Number" to a date or time value, 01/01/2000 12:00 PM, for example, is converted to 36526.5. The value preceding the decimal point corresponds to the date; the value following the decimal point corresponds to the time. If you do not want to see this type of numerical date or time representation, change the number format (date or time) accordingly. To do this, select the cell containing the date or time value, call its context menu and select **Format Cells**. The **Numbers** tab page contains the functions for defining the number format.

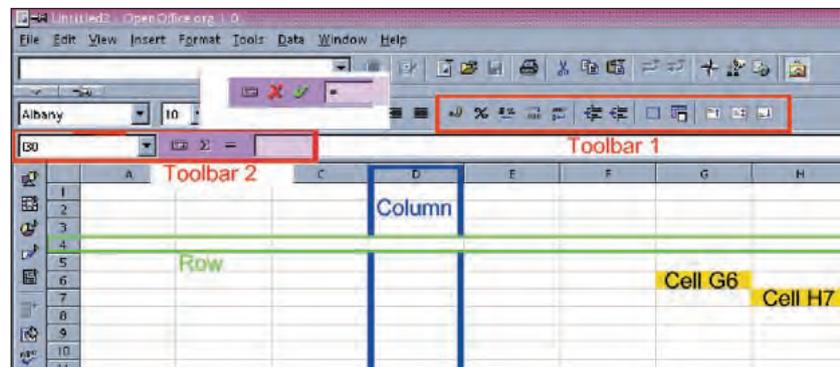
To access this function...

Open AutoPilot: Functions - Category **Date&Time**

A page from the Help files. The picture shows you the kind of detail you get with **OOo**. There are plenty of hyper links, the system is easy to use and you get some nice explanations.

OOo has a comprehensive Help system that is well written and easy to use. It is searchable, you can place bookmarks in it and it has a good success rate. Congratulations to whoever wrote the intelligent behaviour. Unlike the intrusive Help in **MS Office**, **OOo** flashes a small light bulb picture in the bottom right of your display. It is telling you that it has either done something for you (such as autocorrected a mistake) or it

thinks that you might like to know something. It offers you help on file extensions when you first try to save your file, but it is never intrusive. Ignore it and it goes away after a short time. The entries are comprehensive, the only thing I would like to see is a few tutorials included. However, **OOo** is so easy to use it's a minor issue. The level of detail is good, I'd say that it's the best Help system I've encountered in a Linux program.



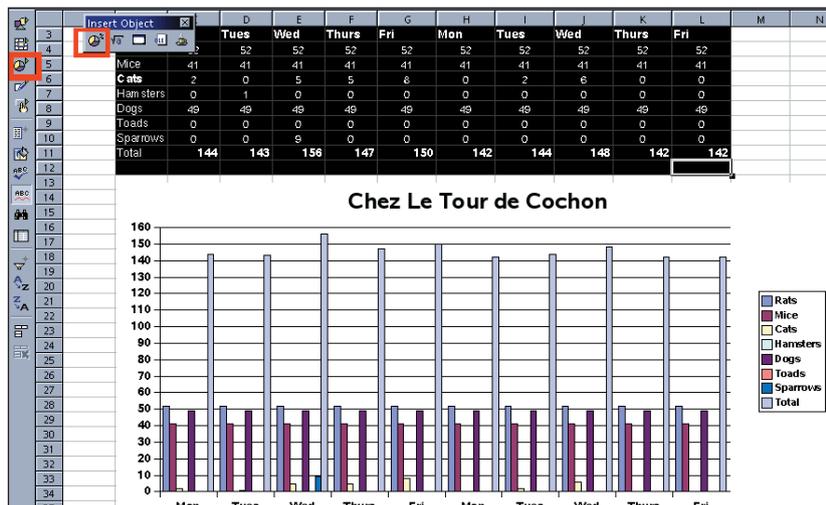
want your answer, then click and drag over the numbers you want to **SUM**. Hit the **Enter** key to confirm. You can display a **SUM** anywhere you like, it does not have to act on the Cells next to it. When you use **SUM** (or other formulas) it changes the Toolbar to show a tick or cross. The tick (or **Enter**) will insert your formula, the cross abandons the operation.

Formula

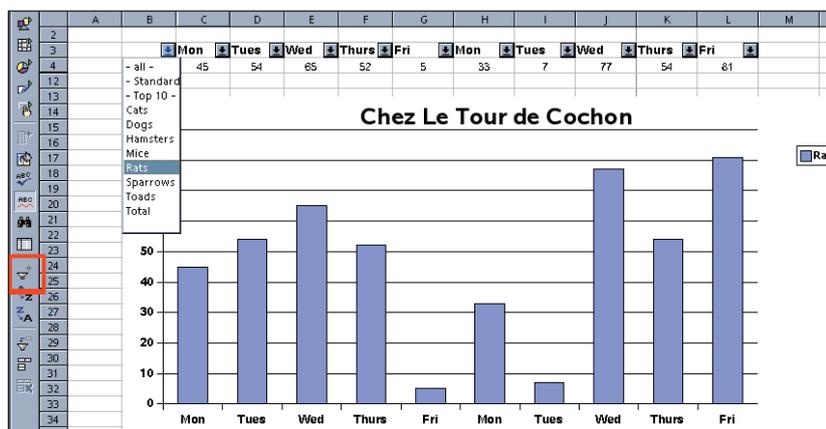
The Equals sign is clicked to tell the spreadsheet that you want it to treat the thing you are typing as a formula and act on it. If you type a command without the equals sign, it will display it a text (which might be what you want to do). The box to the right of the equals sign shows what the formula in a cell is. You can edit it. The cell in the body of the spreadsheet shows the value of the formula.



The **OOo** interface. The alternative **Toolbar 2** display is also shown. Refer to the article for explanations of the various icons.



Graphs (or charts) in OOo Calc. Drag a box around the data, click the icon and follow the wizard. There is no choice of colours, but it does a good job.



The Auto filter can cause just one type of data to be displayed.

Over on the far right of the work area, there's an unobtrusive arrow and push pin. This opens out to act as a formula "quick reference" that you can leave open in the same way that you can with The Gallery (shown concealed in the screen shot above the word Albany). The types of functions are sensibly divided into groups. It gives you a brief description of the syntax underneath each function. If you can't see it, go to Insert>Function List and click to make it visible. You can only get the Function Autopilot from the Insert menu. This launches a wizard. It seemed inconsistent that it did not appear on the File>Autopilot menu.

Graphs (or charts)

OOo makes nice graphs and it's not difficult to alter their appearance. You can change how they look, move the graph around and, by right-clicking, get an Edit menu. One quality I liked was that if you click on a bar in the graph, the cell it refers to on the spreadsheet is highlighted. However, if the graph you made isn't what you wanted, there's not a lot you can do. Just delete the graph and recreate it. Note that if you change any of the values that have gone into the graph, it updates to show the latest data. Sometimes the Auto Calculate behaviour is not what you want. If you have a really big spreadsheet and need to make a lot of changes, it can be annoying to have to wait for everything to be recalculated.

Tools>Cell Contents>Auto Calculate toggles it on and off. F9 will then recalculate the sheet when you are ready.

I have a spreadsheet showing the types of animals eaten each day at the notorious "Chez le Tour de Cochon" restaurant. Although it looks nicer if you leave spaces in the body of your data, OOo will put the spaces into your graph as "Row 4" or "Column B", so get rid of them before you make the graph. OOo uses the names in the top Row and leftmost Column of the area you select as data names. Do not put the graph title in the area you select for the graph. To make a graph, first click and drag a box around the data you want to include (get the Row and Column titles in too). Then either use the "Insert>Chart" menu item or click out the pie chart on the left hand side of the worksheet. The program starts a wizard which helps you make the graph. It's very helpful to tick the "Show text elements preview" box, you'll never get what you want right without it. You can alter the titles and put a heading in before telling it to Create. You then drag out a box and OOo draws your graph.

3D or 2D?

OOo can create various types of graph, although some are not very useful. Think hard about what you are trying to show and choose the graph type that shows the information in the clearest manner. 3D graphs tend to distort the data (they include perspective) and should not be used if you want the people reading the graph to have an accurate understanding of the relationships shown on it.

The Auto filter icon is worth investigating if you use graphs. Perhaps we want to show just the number of rats eaten each day at our restaurant. It can be used to select certain data values from a database shown on the sheet for display. We select the Rats option in the drop down box the graph updates to show us just the number of rats eaten. Be careful not to turn off Auto Filter when a data type is displayed, it can prevent you from going back to the original display. It's probably a good idea to ensure you have saved your graph first.

We can also sort alphabetically (or reverse alphabetically if that is what you want.) Highlight the range of cells to sort, click the A-Z icon and it does it. Remember that if you just highlight the names of things, the data alongside is left unchanged. If you highlight the whole chart, it sorts the names and the data that belongs to the names, but be careful that cells within it that perform any kind of mathematical operation on the contents are

Gnumeric, ergo SUM

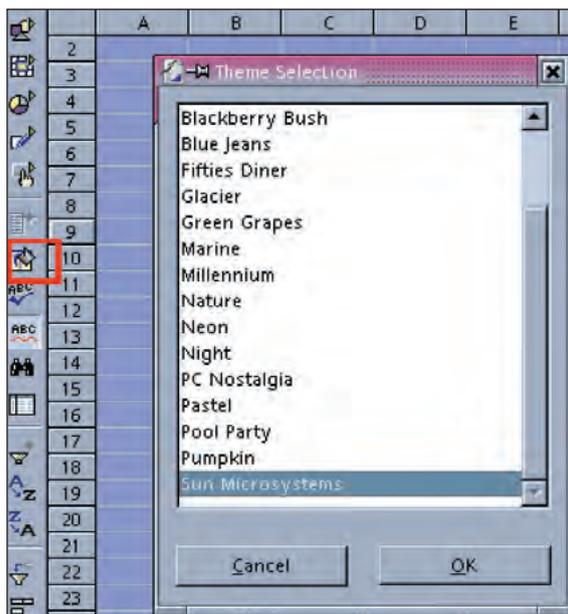
Sun comes through

In the *LXF Gnumeric* article on graphs, I said that I had some problems with the **SUM** feature. I made a simple spreadsheet, saved it as an *Excel* file and imported it into *OOo*. I got an error, it did not like how *Gnumeric* saved the file. I was not sure where the problem was, but Jonathan Mills of Sun UK (who sell *StarOffice*, the commercial version of the code) contacted me via *LXF* and asked for the spreadsheet and looked into it. It seems that the problem lies in *Gnumeric's* code somewhere. There's nothing wrong with *OOo's* **SUM** functions.

What does that tell you about the attitude of the people at Sun who distribute the code? Would Microsoft contact someone who thought they had found a bug in their code or just deny it existed? Welcome to the world of Linux, a genuinely better way of using computers. Many thanks to Sun UK for helping to sort out where the problem was.

	A	B	C	D	E
2					
3			Mon	Tues	Wed
4		Rats	52	52	52
5		Mice	41	41	41
6		Cats	2	0	5
7		Hamsters	0	1	0
8		Dogs		2	0
9		Toads	Dogs	49	49
10		Sparrows	0	1	
11		Total	Mice	41	41
12			Rats	52	52
13			Sparrows	0	0
14			Toads	0	0
15			Total	144	143
16			Mon	Tues	W
17					
18					
19			Total	#REF!	#REF!
20				0	0
21			Sparrows	0	0
22			Rats	52	52
23			Mice	41	41
24			Hamsters	0	1
25			Dogs	49	49
26			Cats	2	0
27			Mon	Tues	

Be careful with the alphabetic sort tools. The red shading shows the original data. The blue is the ascending filter, with the days of the week moved to the bottom. The data is still okay, as "Total" came after "toads". The green one shows errors at the top as the cells no longer add up correctly.



Theme selection. Note how the cells have changed to mid-blue.

not affected. It's best to make sure that the totals, averages etc. are not highlighted.

More useful tools

However, people make mistakes, alter things or paste parts of the spreadsheet into other areas and then wonder why it no longer works. Sometimes you forget what you are trying to do. OOo provides a tool called Detective to help sort out problems. Highlight the cell where you get an error, click on Tools>Detective

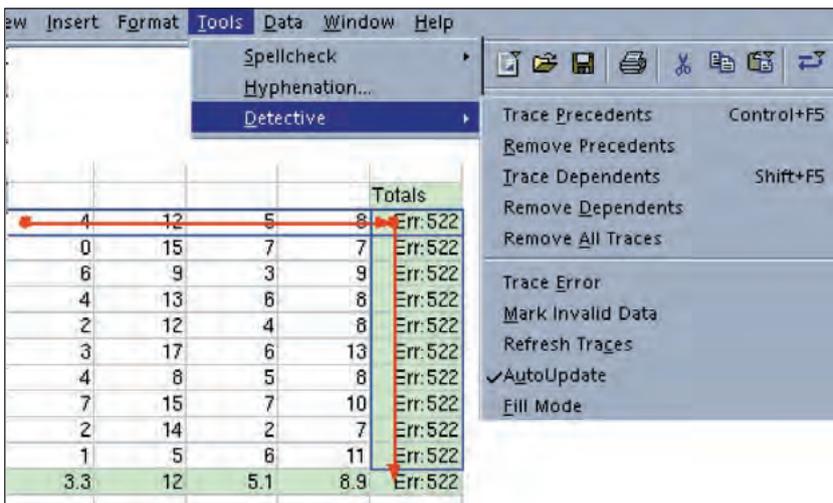
Imports and Exports

The Linux balance of payments

OOo will open and save your spreadsheets in Microsoft Excel 97/2000/XP file formats. I have a few Excel spreadsheets on my Windows partition. Whenever I try to open a spreadsheet with OOo, it always opens it correctly. The only minor problem you might have is that sometimes you don't have the correct font on your Linux machine. I've tried every file I have and never encountered any problems, but that does not mean that you won't have problems. Load OOo (it also runs within Windows) and try it first.

Data is precious, ensure that it loads the files you use. If you have custom spreadsheets at work, try them first, compare the data produced by a Windows Excel sheet to the same sheet in Windows OOo. If it works okay in the Windows version, you should have no worries in Linux.

This means that OOo could be the Linux "killer application". Now you don't have to use Windows to open your Word, PowerPoint and Excel files.



The Detective tool, useful for finding errors in a spreadsheet. Click the cell to find out what other cells contribute data to it.

and either Trace Dependents or Trace Precedents. A red arrow or blue outline box appears to show you which cells contributed data to the mistake. Excel has a similar feature (called Auditing) so it's not original, but if you are familiar with Excel, you'll feel right at home.

The Themes icon (a paint bucket) allows you to apply a colour scheme to your spreadsheet. OOo gives you a few to get going. It's not something you might think of a use for, but if you have lots of spreadsheets to deal with, it's not a bad idea to make the backgrounds of all one type of spreadsheet (perhaps for bonuses earned) all one colour. It might help prevent mistakes.

Conclusions

OOo Calc is a deep program and you could easily fill the whole magazine on this one subject. There's so much to it, it's hard to decide what to leave out. You can make interactive forms (there's a wizard to help), use Java or OLE objects, I could spend pages discussing Functions, but the Help files give you enough advice. If you want a replacement for Excel, OOo Calc has all the tools you need. If you need to reduce costs for software licences, OOo is free, install it on as many machines as you please. If you are worried that the company that provides your present software is just treating you as a source of income and you don't really want to have all your data tied in a deliberately incompatible format, then OOo could be the solution you need. I've not used the Windows version of OOo, but if I relied on selling MS Office for a living, I wouldn't be taking out any new bank loans. [LXF](#)

TEMPLATES

Practical PHP programming



This month, **Paul Hudson** explains how template systems allow you to separate site design from site content.

The original idea behind templating is fairly simple – if you abstract the code that *displays* content from the content itself, the end result is a cleaner, more simple site. This was the same thinking behind Cascading Style Sheets (CSS), in that CSS enabled you to take all the formatting and layout code embedded inside your HTML and place it into a separate file.

PHP itself was, in a way, just a template system at the very beginning of its life; its main purpose was to allow you create various types of content (guestbooks and other such web regulars) using special keywords and a limited command set.

Now that PHP is a full language in its own right, things have gone full circle – you can now create a template system in the language that used to be just a template system itself! In this article, we'll be looking at how templates work by creating our own very basic templating system. Then, once we're through with the theory, we'll be taking a look at the four most popular template systems already written for PHP, and seeing how they compare against each other.

Getting started with templates

The theory behind a basic template system is quite straightforward. A PHP developer, like yourself, creates a variety of flexible scripts to do various common jobs on your web site. These scripts, called templates, can be inserted by designers into HTML pages through the use of custom tags. During execution, these tags are replaced with the appropriate PHP scripts to form a complete page, and the effect is transparent to the end user.

It's important to remember that templating systems are designed to make life easier for everyone. For the programmer, having a template system means they can set up the templates once and not really have to come back to them unless a serious change to the site design has been made. For the web designers (whom we will assume are non-programmers; think *Dreamweaver*), it means they have the ability to make use pre-written code without having to worry about whether it will work or not. That is, they can do that if you have done your part of the job properly!

Here's an example of how this would work, using a page that, once the appropriate templates were written, could be created by our non-technical designer. Save it as `welcome.html`:

```
[HEADER]
```

```
Welcome to BlahdeBlah Ltd. Please fill out the following form to sign up for our regular email newsletter.
```

```
[NEWSLETTERSIGNUP]
```

```
[FOOTER]
```

As you can see, it's all plain text – not a `<?php` or a `?>` in sight. Instead, keywords are used, encased in square brackets in order to differentiate them from plain text, to represent where PHP code should go. The designer sprinkles the special keywords into their layout wherever they want pre-written functionality to appear, and their job is done – no programming required.

To turn the designer-produced keyword/layout mix into a useable page, a simple parser would have to be written that locates each keyword and converts it to the appropriate PHP code. So, `[HEADER]` is turned into the full page header for the site, `[NEWSLETTERSIGNUP]` into a form, etc.

Let's take a look at the code necessary to convert this into a page that can be shown to visitors. Naturally, we cannot let people link directly to the HTML page because it will not be parsed by PHP, so we need to write a master script to handle all page requests.

Here's the code for an example master script; save it as "default.php":

```
<?php
if (!$page) $page = 'default.html';

$page = "$page";
$fp = fopen($page,"r");
$content = fread($fp, filesize($page));
fclose($fp);

$content = str_replace('[HEADER]', '<?php include
\'header.php\'; ?>', $content);
$content = str_replace('[FOOTER]', '<?php include
\'footer.php\'; ?>', $content);
$content = str_replace('[NEWSLETTERSIGNUP]', '<?php
include
\'newsletter.php\'; ?>', $content);
eval('?' . $content);
?>
```

That's all the code necessary, so it's nothing *too* complicated.

Firstly, we check whether we have a page variable set. If we don't, we set a default page to load, which is 'default.html'. Make sure you have your `DirectoryIndex Apache` directive configured to load `default.php` before everything else, otherwise, if you opt to put the unparsed HTML in the same directory as the PHP parsing script, *Apache* may serve

up index.html before your parsing script.

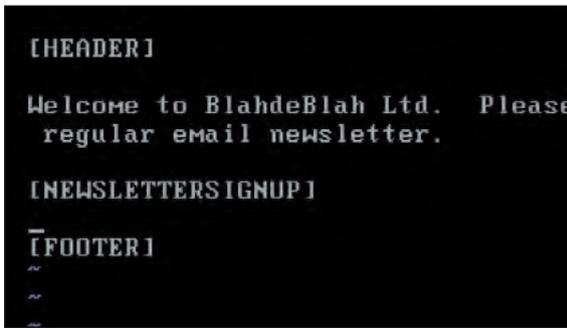
We then load in the page that was requested, using the standard **fopen()/fread()/fclose()** code we looked at back in *LXF31*. I recommend you don't put the content pages in the same directory as the template code itself. There's no harm keeping all the files, except default.php, outside the public_html directory entirely, in fact I would recommend it – it's the easiest way to ensure hackers aren't unable to dig up more information than you want. On that note, you may want to filter the value of **\$page** so that people can't provide a value like **.././../etc/passwd** – this is a very common trick, and I have seen it work on many sites that use the **include()** function. The simplest way to solve this is simply to reject **\$page** if it doesn't start with one directory above your root public HTML folder, e.g. **/home/paul/www/**.

Back to the parsing script, the continues on by replacing all our template keywords with PHP **include()**s leading to our pre-written code. For extra speed, you can just directly write the code itself rather than calling **include()**, but space is at a premium here, so **include()** is preferable. Whilst **include()** is a touch slower than putting the code directly into the page, it keeps the master script easy to read, which is probably a bigger advantage in the long run.

Finally, we call the **eval()** function, passing in our newly-parsed content prefixed with a **?>**. **eval()** executes PHP code as if it were a script by itself, which is immensely helpful because it allows you to create your PHP code dynamically at run-time, as we do here. The reason we prefix our content with **?>** is because **eval()** is expecting PHP code. We need to start with a "drop out of PHP mode" command because it is very unlikely that our non-technical web designer will have started their content with PHP code, or, if they need to, they should at least start their script with **<?>**. The worst case is that they start with a template command, which means the first thing **eval()** will encounter will be the **?>** we prefixed to the content followed by the **<?>** of one of our includes – hardly anything to worry about.

So, with just nine lines of code, we have the most basic templating system ready. Of course, you'd still need to go ahead and write all the individual templates (e.g. **HEADER**, **FOOTER**, etc.), but those can be added over time – our non-technical web designers can start using the system with just a couple of templates, with more being added later.

Just to get you started, here is the code I would put into header.php, based upon the operations I perform on most of my pages – you are welcome, of course, to insert your own code!



Editing the template is easy for all – no PHP in sight!

```
<?php
mysql_connect("localhost", "username", "password");
mysql_select_db("lxfdb");

session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>Welcome to BlahdeBlah Ltd!</TITLE>
</HEAD>
<BODY>
```

In order for our templates to be loaded correctly into the HTML, all our pages would need to be loaded as something like default.php?page=index.html, and linked to accordingly. This somewhat clumsy requirement is good for beginners as it requires little to no implementation effort, however a much better option (if somewhat trickier to implement) is to have your web designers save all their files in your public HTML directory with a .php extension, then use the **auto_prepend_file** directive of your *php.ini* file to specify the location of your parser. This only works, of course, if you want *all* your PHP pages to have your parser attached to it.

Once you've filled in nlsignup.php and footer.php – you can use blank files for now, just to get started – try loading the URL default.php?page=welcome.html into your browser; all being well, you should see the completed system in action.

Once working, even in this basic format, you will easily see the power of templates – many companies have designers and developers working together to produce one page. Implementing templates, even in the simple example shown above, can



Adding more to your template engine

For those who like to roll their own

If, like me, you want maximum performance when using templates, the only real option is to write it yourself. By doing so, you get to decide precisely which features you want to include and which not, and that can easily turn 50 requests per second into 100. A very easy feature to add which can make quite a difference to speed is template caching. Harking back to my PHP accelerators head-to-head again, you'll remember that the basic action of all accelerators was to cache compiled PHP scripts in order to speed up execution. The caching of templates is entirely separate to PHP compiling, and as such they work together very well, providing a joint speed boost. Template caching works like this:

- i) Receive URL like default.php?page=index.html
- ii) Grab the modified date of index.html, and compare it to the modified date of ./cache/index.php.
- iii) If index.html is dated older than the cached copy, then simply **include()** the cached copy.
- iv) If index.html is newer than the cached copy, we need to re-generate the page. So, run all our **strreplace()**s, echo out the output, then save the contents to the cache.

Simple, but effective. As implemented above, the master page will usually serve the cached page, which is much faster than having to parse all the

text every time. Only if the template page is changed by our web designer the cached page be re-generated. This means that the hard work of all the **strreplace()** calls is only done once every few hundred (thousand?) requests.

From there, PHP accelerators will cache the cached files, if that makes sense. So ./cache/index.php, which is the file generated from the template so the master script doesn't need to keep re-parsing the template, will itself be held in the RAM cache by the PHP accelerator you're using. This is what provides the double speed boost, and is very effective in real-world situations.

TutorialPHP

« eliminate hours of to-ing and fro-ing between artist and programmer by transferring more control to the designers.

Given the flexibility offered with such a simple system, you might ask, "Where's the benefit in using a more complicated system?" If you read my PHP accelerators head-to-head article in *LXF34*, you will have noticed that, even when accelerated, the *Smarty* templating system managed to output only ten requests per second on a reasonably powerful (albeit absolutely unoptimised) Debian box – it must offer quite a lot to bring about such slow performance!

However, slow as it may be on occasion, clearly templating is popular with the masses because there is a growing collection of various solutions available to the programmer interested in something off the shelf. For the remainder of this article, I'm going to briefly look at the four most popular template solutions available for PHP (*Smarty*, *TemplatePower*, *FastTemplate*, and *PHPLIB*) briefly highlighting what each offers and roughly how it performs.

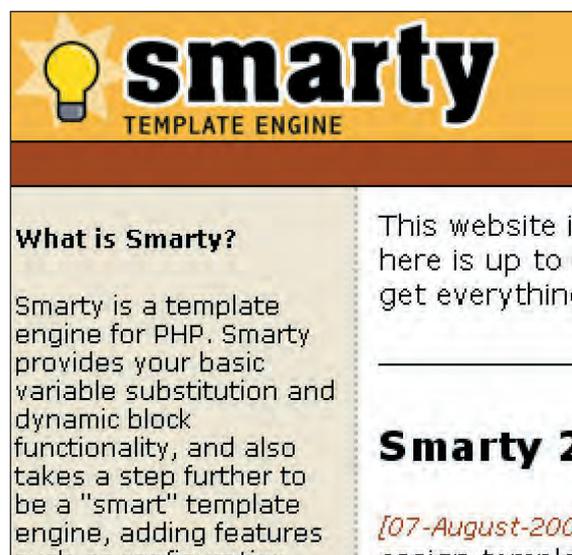
Smarty

In recent times, the *Smarty* template system has grown in popularity immensely, and this may well be due to the fact that the feature set has grown immensely also! Created and maintained by four key members of the PHP community, and having been in development since late 1999, the driving principle behind *Smarty* is that developers shouldn't ever need to see HTML at all – designers should be able to do as much as they can.

As with all template systems, *Smarty* handles basic keyword replacement. However, that is the least of its functionality – *Smarty*'s features include simple string operations such as uppercasing and lowercasing, as well as more powerful features such as running regular expressions. Also, with a little training, our non-technical web designers can be using if statements inside their HTML, as well as loops.

The end result of all this is that *Smarty* is a miniature programming language of its own, with wordy statements that make programming a little bit more guessable. For example, to create a drop-down **SELECT** box where visitors can select a year, our designer would need to enter `{html_select_date start_year=1998 end_year=2010}`. Through a clever combination with the ever-popular *overLIB* JavaScript library,

The *Smarty* homepage is full of great user information, and is well worth a read.



Why not use templates?

Well, they're not for *everyone*...

Templates, whilst being very helpful, certainly are not suitable for all environments. If you find that more often than not the pages you create are very specific to a certain task – that is, they'd be hard to generalise into a template-like form – then you definitely fall into the category of programmers who wouldn't get much benefit from switching to templates. Sometimes it is even the case that switching to templates can create *more* work for you as a web developer – the act of switching to templates might end up with you spending all your time adding more complexity to your templates, creating various exceptions for specific cases in your sites. We all want to avoid as much programming pain as possible, so if in doubt, try phasing templates in bit by bit.

Another very important reason to avoid using templates is that you lose direct control of the code in quite a large way, at least if you intend to use an off-the-shelf system. Recently, I read a php-dev mailing list thread where *IMP* for the *HORDE* code framework was apparently crashing the *Zend* engine. If I had implemented this and it was crashing on my box, I'd certainly be unhappy – partly because it was crashing, and partly because, as the code isn't mine, I wouldn't be able to fix it! Losing control of the situation is always a bad thing, so, if you definitely want to switch to a templating system, at least use one that is stable.

Smarty also includes a "popup" function that automatically generates an *overLIB* tooltip, complete with all the customisation you could need.

It's fairly easy to argue that in some ways *Smarty* offers too much – an experienced web designer using *Smarty* probably knows more than enough to program quite a lot in PHP themselves. Naturally, the jump from using templates to *programming* them all depends on the complexity of your templates! However, many people treat *Smarty* like a graphical development environment. That is, while it's *possible* to create a GUI program from scratch using just code, many people prefer (perhaps quite rightly) to use tools like *Kylix* or *Qt Designer* to make use of as much pre-written functionality as possible.

Adding all this functionality to an interpreted language, however, comes at a hefty cost: your basic *Smarty* installation runs like a dog. With a limp. By itself, the one mildly redeeming factor is that *Smarty* has special caching functionality to alleviate the speed problem, and it certainly does make quite a difference. The key to this feature is that *Smarty* stores away its processed template in a cache, and this is what it will use if a request comes in for a template page that has not changed since it was last parsed.

For even more increased performance, the auto-update check can be disabled, as long as you remember to refresh the cache when you make a change. However, this all changes once a PHP accelerator (e.g. *Zend Accelerator*, or *ionCube's PHP Accelerator*) is installed. In some *Smarty* scripts, you can expect an increase of five to ten times that of the unaccelerated code, which pushes it above and beyond the best efforts of other template systems.

One extra feature that is entirely unique to *Smarty* is its excellent plug-in architecture, which allows you to add your own code to extend the system. This is a very smart move on behalf of the *Smarty* team (no pun intended!), and one I hope is soon seen in other template systems. All in all, it is features like the caching and the plug-in system that help bring *Smarty* up to a

very useable level of performance, although harbour no doubts – it is often still sluggish if you're not careful.

TemplatePower

TemplatePower was a long-running favourite amongst developers until recent times, when the script performance started degrading beyond what was considered workable. It seems that the developers of the software opted to keep adding features rather than make the existing ones work in a more optimised manner – a great thing if you always want to stay on the cutting edge, but the added functionality hits performance quite hard now and then. Still, you will generally find that unaccelerated

TemplatePower outperforms unaccelerated *Smarty* at pretty much every test excluding tight code loops, and it certainly has enough features to distinguish itself as a very powerful template system – template caching, file includes, and the like. Whilst raw performance is definitely the highest point of *TemplatePower*, the software development rate drags the system down a little. With *Smarty* being written by such a tightly-knit team of very productive developers, it certainly has quite a programming boost over the other template solutions and this shows, if nothing else, in the sheer speed-up seen by users upgrading from 1.5.x *Smarty* to 2.x.

Expect *TemplatePower* to perform well enough in all conditions; whether unaccelerated or accelerated, it makes a nice middle choice between power and speed. However, with *Smarty* development continuing at its current pace, it won't be long before *Smarty* takes over on both accounts!

FastTemplate

As one of the oldest template systems available for PHP, *FastTemplate* certainly has a very mature codebase. For a start, it was ported direct from the original Perl template script, and so many people were able to transfer across to it with little trouble. Of course, for some, having been ported from Perl is a *bad* thing. No comment ;-) Other than having been around a long time, *FastTemplate* sadly doesn't have much going for it. The functionality provided is much more simplistic, sticking pretty much to the most basic type of templating – pattern matching and replacing. Also, "FastTemplate" is a bit of a misnomer; don't expect it to set any new speed records. Also, don't expect regular updates – or, indeed, any updates! Since the templating scene became more popular, *FastTemplate* has pretty much died out, and is generally only used as a legacy system. Some people have tried to build on the solid codebase by adding functionality normally only seen in newer systems, with *Cached FastTemplate* (written by Jesus Castagnetto, co-author of the excellent book, *Professional PHP Programming*) giving quite a solid speed improvement. However, it's perhaps a case of too little, too late. There are only two reasons I can see people taking an interest in *FastTemplate*, and they are:

- i) To give themselves a headstart towards writing their own template system.
- ii) Because it's cross-language, which makes porting code to/from Perl, Python, or even ASP easier.

If you're looking for a templating system and don't think either of those reasons are appropriate for you, I suggest you try another solution instead.

PHPLIB

At its peak, *PHPLIB* was a saviour for many PHP programmers, as it introduced functionality like per-session variables, easy to

```

Smarty Debug Console
included templates & config files (load time in seconds):
index.tpl (0.07684) (total)
test.conf [setup] local (0.00022)
header.tpl (0.00536)
footer.tpl (0.00202)
assigned template variables:
{$Class}
Array (4)
0 => Array (4)
0 => A
1 => B
2 => C
3 => D
1 => Array (4)

```

implement user authentication and templates, which were nowhere else to be seen in the PHP world. While *PHPLIB* no longer seems to be actively maintained, the template implementation is very stable in its current incarnation.

Although *PHPLIB* was the original template system written specifically for PHP, a distinct downside to it is that you spend a great deal more time learning the template code, which makes the learning curve perhaps too high for most non-technical designers. When using *Smarty*, you can often pretty much guess a lot of the code required because it's almost a natural language system, but this is rarely if ever the case when programming with *PHPLIB*. Furthermore, the main reason I stopped using *PHPLIB* templates myself was because it required a great deal more work to create the templates themselves, because they were often very specific to a certain situation. This is, of course, the bane of templates!

On the performance front, *PHPLIB* does surprisingly well for such an old system, generally pulling ahead of *TemplatePower* on the whole. Whilst it's important to remember that benchmarking with templates is almost impossible to do in an unbiased way simply because the scripts have to be re-written for each system, I encourage you to try speed tests with *PHPLIB* templates simply because I've had so many people recommend it to me in the past.

Conclusion

Even though this primer has been quite short, I hope you now at least have an idea as to how a templating system can speed up workflow in most development environments simply by letting designers design and developers develop. When I first started writing PHP, a lot of my work was just writing the same things again and again – "Paul, could you make a form that...", "Paul, could you write a messageboard where...". I realised that after a while, I wasn't really developing anything at all, merely re-writing what I had done before. Nowadays, I myself use templates so that I can develop something once, then pass it onto the designers I work with to use in their pages, which lets me get onto the challenging (and fun!) tasks. If you have ever been in the same situation, then templates are for you. You may not need all the power and flexibility that comes with a pre-built template system, but at least consider writing one to fit your own needs.

Does "only *Smarty* have the answer"? It very much depends on how you want to implement your templates. As I've said already, the most important thing to remember is that templates are supposed to development easier, and that should be the golden rule of all would-be templaters. Don't let yourself fall into the trap of producing a template system so complex that it ends up being a productivity vortex! 

Smarty has a very cool debug console built in to help you locate and solve problems in your code.

About Paul Hudson

Paul Hudson is a London-based web developer specialising in PHP and Perl. He can be emailed at hudzilla@php.net.

NEXT MONTH

Next month we'll be starting a three-part mini-series looking at dynamic media generation using PHP. In part one, we'll look at how to generate images dynamically using code, including how to resize and resample existing images; in part two we'll move on to how to generate animated Flash movies, complete with buttons and actions; and in part three you can expect extra goodies on how to create printable PDF files. If you have any comments or suggestions about this series, please be sure to write in!

SYSTEM ADMINISTRATION

Process Accounting in Linux

Keith Gilbertson keeps tabs on where all those CPU cycles are being spent.

While onsite at a fortune 500 corporation recently, I overheard a tech support person directing an excitedly whispered warning to a project manager. "Don't play any games on your PC! The corporate auditors have a way to find out exactly what programs you use, and for how long!"

After loudly assuring the techie that he was all business and did not intend to play games anyway, the manager smiled and said in a much quieter tone that he needn't be concerned; he was using Linux, and not Windows like most of the rest of the company.

If the tech's tale is true, the manager may indeed have reason for concern. While the rumoured auditing app at this particular company was developed for Windows, the Linux kernel has a built in process accounting (PA) facility that allows system administrators to collect detailed information in a log file each time a program is executed on a Linux system. With this capability, our mythical evil corporate auditor could in fact collect information about who has been playing games on a computer and for how long.

Although a company's interest in knowing which employees have been playing *Solitaire* on company equipment is of questionable utility, there are in fact good reasons to use process accounting. In this article, I'll discuss some situations where process accounting is useful, explain where to obtain and how to use the standard process accounting commands, and then demonstrate how to use the process accounting structure and system call in C programs.

Preliminaries

I've assumed that your system has process accounting support compiled into the kernel. I make this assumption because the kernels on all of the Linux systems that I have access to are configured to allow process accounting, but your distro may be different. If you compile and run the first code listing in this article as root with no command line arguments but receive an error message, it is likely that process accounting support was not included in your kernel. You'll need to compile a new kernel and answer yes to **CONFIG_BSD_PROCESS_ACCOUNTING**. Though recompiling your kernel is beyond the scope of this article, instructions can be found at the Linux Documentation Project.

Keep in mind that on busy systems, turning on process accounting can use significant disk space. On my Pentium III system with RH7.2, each time a program is executed 64 bytes of data are written to the process accounting log file. While researching this article and running the process accounting utilities on a test machine with a low disk space, I discovered a monitor process that

executes every second. The drive on this machine filled up quickly. Some servers' daemons will initiate a separate process for each incoming connection. On a production server which spawns nearly 25,000 processes per hour, approximately 1.1 gigabytes of process accounting data are generated each month. Utilities such as the *accttrim* and *handleacct.sh* script listed in Table 1 are available to truncate, back up and compress log files at regular intervals. If you plan on doing process accounting on a busy system, it will be important for you to learn about and use these utilities.

Finally, know that you must have root privileges on your Linux system to enable or disable process accounting, whether using the standard commands or creating your own.

Uses of process accounting

One of the earliest uses of process accounting was to calculate CPU time absorbed by users at computer installations that were expensive to maintain, and to bill the users accordingly. With the greater abundance and relatively low expense of today's computing resources, this application has fallen by the wayside. It's possible that this application could again become important, though, if the distributed computing model catches on.

Sysadmins may wish to use data collected from PA facilities to monitor which programs are accessed the most by users, and then optimise the system configuration for these types of programs. *E.g.*, part of the data collected by the PA facilities includes the number of bytes input/output by the program, and also the CPU usage. A system that runs a high percentage of i/o-intensive applications may need to be optimised in ways that a system running a high percentage of CPU bound applications should not be optimised.

An administrator might sometime be required to evaluate two products with similar functionality. Lets imagine that before making a selection, the administrator wishes to see which fish forecasting product the users of the system are actually using. Process accounting can be turned on for a week to record the names of all commands executed into a logfile. The administrator can then parse the logfile to find out which command was run more often, *FishFind* or *Hookem*.

The most typical applications of process accounting are as a supplement to system security measures. In the case of a break-in to a company server, the log files created by the process accounting facility are useful for collecting forensic evidence. A careful look at the programs an attacker has used on the compromised system can provide useful information about the damage the attacker has done, as well as the intruder's methods and possible motivations. Evidence collected from the process accounting logs may also be helpful in court. I know of one criminal case in which this data, when uncontested by the defendant, was enough to produce a misdemeanor conviction.

Standard commands

Even if the PA facilities have been compiled into your kernel, you might not have the user commands for process accounting installed on your system. If this is the case and you're looking to get started quickly, you should first try finding the PA commands for your

specific Linux distro. The package for your distribution is likely to be configured to place log files in the appropriate location for your system's setup. This will make installation much simpler. On my RH7.2 distribution C.D.s, I found the `ps-acct-6.3.2-9.i386.rpm` on the second disc in the `/mnt/cdrom/RedHat/RPMS/` directory. If you use the `gnorpm` graphical install tool, the package will appear in the hierarchy at `Packages/Applications/System`.

Depending on the version of the utilities that you install, the programs could be licensed under the BSD license. The BSD version of the utilities are based on an early implementation. These utilities are available by clicking here. The filename will be similar to `acct-1.3.73.tar.gz`, with small differences depending on the version number. In order to get these utilities to compile on my system, I had to edit the `lastcomm.c` file and comment out the prototype for the `strcpy` function.

There is also a process accounting utilities set that is licensed under the GNU GPL. For those who prefer the GNU GPL, or for those simply unable to find the utilities for their distro, Noel Cragg's *GNU Accounting Utilities* are available from the GNU website.

The exact process accounting commands that will be installed on your system will vary depending on the particular package you've chosen. The table below shows a list of the process accounting commands that you could encounter and the purpose of each:

Table 1 PROCESS ACCOUNTING COMMANDS

COMMAND NAME	PURPOSE
<code>accton</code>	Enables or disables process accounting
<code>acctentries</code>	Counts the number of accounting entries in the log file
<code>accttrim</code>	Truncates the accounting file specified
<code>dumpacct</code>	Dumps the contents of the log file
<code>dump-acct</code>	Similar to <code>dumpacct</code>
<code>handleacct.sh</code>	Script to compress and backup logs and delete the oldest
<code>lastcomm</code>	Prints commands executed on the system, most recent first
<code>sa</code>	Summarise accounting information

Installing the GNU utilities

After you have downloaded the source for the GNU Accounting Utilities into your home directory, you should open a terminal window and type the following commands:

```
tar zxvf acct_6.3.5.orig.tar.gz
cd acct-6.3.5
./configure
make
su
(Enter the root password when prompted)
make install
```

A few basic process accounting commands have been installed on your system. You're now ready to turn on the accounting and start using the commands.

In this brief introduction to using the process accounting commands, I'll look at just two commands: `accton` and `lastcomm`. I've chosen these two commands because they are standard and included with all versions of the utilities.

Accton

The `accton` command switches process accounting on or off. If a filename is specified on the command line, that filename will be

Resources

Web or paper

The Process Accounting Mini-HOWTO

Albert M.C. Tam

www.tldp.org/HOWTO/mini/Process-Accounting/index.html

Linux Documentation Project Kernel-HOWTO

Brian Ward, Al Dev (Alavoor Vasudevan)

www.tldp.org/HOWTO/Kernel-HOWTO.html

GNU Accounting Utilities

Noel Cragg

www.gnu.org/directory/System_administration/Monitoring/acct.html

Other process accounting utilities

www.ibiblio.org/pub/Linux/system/admin/accounts/

Linux System Security

Scott Mann and Ellen L Mitchell

Prentice Hall PTR 2002

used to log the process accounting information. If no argument is specified, the process accounting will be switched off.

To start the process accounting facilities on your system, perform an `su` to become the super-user. Make sure that the logfile exists by performing a `touch` on the desired location.

Example: `touch /var/log/pacct` Then type the full path to your `accton` program (usually `/usr/sbin/accton` or `/sbin/accton`) followed by the filename. Example: `/sbin/accton /var/log/pacct`

You've just started the process accounting facilities. Here's a note of interest: the data is not actually added to the file when each process begins execution; it is written when a process exits. The aforementioned project manager can play the `xbill` game all day long and not have this information written to the process accounting file, as long as he never exits the program. When he goes home at night, he can choose to leave `xbill` running and minimise the window, or simply power off his computer without performing a proper shutdown.

Now that you've switched on the accounting, you can run a few commands as an ordinary user to get some data for the `lastcomm` command, which you'll use next. When you're finished, `su` to become the super-user once more and run `/usr/sbin/accton` or `/sbin/accton` with no arguments to switch off process accounting again.

Lastcomm

The `lastcomm` command will print information that is contained in the accounting log files, with the most recent record printed first. You can use the `-f` command line flag to specify a filename. Typically, the process accounting logfile on a system will be setup so that only super-user can read it. This command will then be executed by root. Example:

```
lastcomm -f /var/log/pacct
```

If you type in the above command, you will see output similar to the following:

```
id      root      stdin    0.00 secs  Mon Jul 22 12:41
xauth S root      stdin    0.00 secs  Mon Jul 22 12:41
xauth S keithg   stdin    0.00 secs  Mon Jul 22 12:41
xauth S keithg   stdin    0.01 secs  Mon Jul 22 12:41
bubbles X keithg   ??       0.01 secs  Mon Jul 22 12:33
ls      keithg   ??       0.01 secs  Mon Jul 22 12:26 >>>
```

Tutorial Process Accounting

```

<< bash X keithg ?? 0.03 secs Mon Jul 22 08:25

```

Lastcomm will display the command name, flags, username, terminal, and exit time for each command. A particular command, user, or terminal can also be specified on the command line. For example, if you only want to find instances of when the *su* program was started, you can type:

```
lastcomm -f /var/log/pacct --command su
```

Now you'll see output like this:

```

su      root  ??    0.01 secs Mon Jul 22 10:52
su      keithg stdout 0.05 secs Mon Jul 22 09:32
su      keithg stdout 0.04 secs Mon Jul 22 09:17
su      keithg stdout 0.00 secs Mon Jul 22 09:17
su      root  ??    0.01 secs Mon Jul 22 08:04
su      root  ??    0.01 secs Mon Jul 22 04:04
su      root  ??    0.00 secs Mon Jul 22 03:29
su      keithg tty1 0.00 secs Sun Jul 21 19:49

```

Notice that on each line, the command listed in the left column is now "su".

For more details about these commands and the other programs in the table, see the respective man pages.

Programming details

The *acct* structure for collecting process accounting details is documented in the header files */usr/include/linux/acct.h* and */usr/include/sys/acct.h*.

Table 2 MEMBERS AVAILABLE IN THE ACCT STRUCT.

MEMBER NAME	TYPE	DESCRIPTION
<i>ac_flag</i>	char	Special flags showing process behaviours
<i>ac_uid</i>	u_int16_t	User ID of the process
<i>ac_gid</i>	u_int16_t	Group ID of the process
<i>ac_tty</i>	u_int16_t	Controlling terminal of the process
<i>ac_btime</i>	u_int32_t	Process start time
<i>ac_untime</i>	comp_t	User time
<i>ac_stime</i>	comp_t	System time
<i>ac_etime</i>	comp_t	Elapsed time
<i>ac_mem</i>	comp_t	Average memory usage
<i>ac_io</i>	comp_t	Characters transferred
<i>ac_rw</i>	comp_t	Blocks read or written
<i>ac_minflt</i>	comp_t	Minor pagefaults
<i>ac_majflt</i>	comp_t	Major pagefaults
<i>ac_swaps</i>	comp_t	Number of swaps
<i>ac_exitcode</i>	u_int32_t	Process exitcode
<i>ac_comm</i>	char[]	First 16 characters of command name
<i>ac_pad</i>	char[]	Padding bytes

As you can see from the table, there is a lot of information packed into the 64 byte accounting record. This should be sufficient for most uses. If you feel you need more information than is available with standard process accounting, then look at the Mann and Mitchell book listed in the *Resources* box. The authors include a brief description of *auditd*, a tool which records still more information about executing processes.

Example programs

Listing One (Enabling and Disabling Accounting to a File)

```

/* pa.c
 * Linux demonstration program.
 * Logs process accounting information to a
 * file specified on the command line.
 * If no filename is specified, process
 * accounting is switched off.

```

```

*/
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int
main (int argc, char **argv)
{
    int rc;

    if (argc == 1) /* No arguments - switch off */
    {
        printf("Turning off process accounting.\n");
        if ( (rc = acct (NULL)) )
        {
            if (errno == ENOSYS)
            {
                printf
                ("It appears your kernel does not
                " include accounting support\n");
            }
            perror("Problem turning off accounting");
            return rc;
        }
    }

    else /* cmd line arg - switch accounting on */
    {
        printf
        ("Attempting to log to file %s.\n",
        argv[1]);
        rc =
        creat (argv[1],
        S_IRWXU | S_IRGRP | S_IROTH);

        if (rc == -1)
        {
            perror("Problem creating log file");
            return rc;
        }

        if ( (rc = acct (argv[1])) )
        {
            perror("Problem in acct() call");
            return rc;
        }

        return 0;
    }
}

```

The first listing is very simple. It is intended to demonstrate the use of the *acct* system call. The *acct* call takes one argument: the name of the file to which process accounting information will be appended. If the argument is **NULL**, process accounting will

be turned off. The file must already exist when the system call is made, or the call will fail and an error will be returned.

If a program running with the ID of an ordinary user makes a call to *acct*, the call will also fail and return an error. Programs that attempt to switch process accounting on and off must have super-user privileges to succeed.

The code is similar to a typical implementation of the *accton* command. There are two main differences. The first is that this code will report what it is doing with messages to standard output. The second is that if the file specified on the command line does not exist, it will be created.

The file includes the `<unistd.h>` header file. All programs that make use of the *acct* call should include this file. The program checks to see if **argc** is equal to one, meaning no arguments were passed on the command line. If this is the case, the program attempts to turn off process accounting by calling *acct* with a **NULL** argument.

If the command is run with an argument, the program will assume that the first argument is the filename and, if the file does not exist, will attempt to create it with the **creat** system call. Then, the program will call *acct* with the filename as an argument to turn on process accounting. If an error code is returned from a system call, a message will be printed and the program will exit.

Listing Two (Parsing the Accounting file)

```
/*
 * parea.c: Linux program to demonstrate reading a
 * process accounting record into memory.
 */

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/acct.h>

int
main (int argc, char *argv[])
{

    int fd;
    int bytesread;

    struct acct a; /* accounting record */

    if (argc == 1)
    {
        printf("You must supply a filename"
            " on the cmd line\n");
        return -1;
    }

    fd = open (argv[1], O_RDONLY);
    if (fd == -1)
    {
        perror ("Problem opening specified file");
        return -1;
    }
}
```

```
/*
 * Read and print command name from each record
 * in the file
 */
while ((bytesread =
    read (fd, &a, sizeof (struct acct))) > 0)
{
    printf ("%s\n", a.ac_comm);
}
return 0;
}
```

The second code example demonstrates how to read records from the log file into an *acct* structure in memory so that the information can be printed out or operated upon.

This program includes the `<sys/acct.h>` header file. All programs that need to work with the *acct* structure should include this file. Local variables in the main function include a file descriptor, a variable to hold the number of bytes read from the file, and an *acct* struct.

The user of the program must specify a filename on the command line. The program attempts to open this file for read only access. If the open was successful, the program will **read()** a record from the file directly into the local *acct* structure, *a*. Due to space constraints for the article, I've made the assumption that a **read()** will always return exactly the number of bytes requested until the end of file is reached. The program continues to read and print the command name from the records until a zero is returned from the **read()** call, signalling the end of file condition.

The listings in this article are intended only as a simple introduction to the system accounting structures. Robust programs would create a buffer to read multiple accounting records at once, and would check for issues such as fewer bytes read from the file than were requested. To see examples of robust programs, look at the source for the process accounting utilities that you've installed. A huge benefit of Linux and the open source model in general is the high availability of good programming examples.

Conclusion

You now have enough information to enable process accounting and use the standard commands to retrieve information about programs executed on a Linux system. If you're so inclined, you can also learn to make your own custom tools to parse the process accounting log files.

If you're using process accounting for system security, keep in mind that it is not by any means a comprehensive solution, but only one small tool. In fact, as Mann and Mitchell point out, you should be careful about trusting the information in the process accounting log files; the logs may have been modified by a technically-savvy attacker. If you're just starting a system security program, you should start by reading a book, such as Mann and Mitchell, that serves as a broad introduction to the topic.

With a basic understanding of the process accounting tools in Linux, and with some experimentation, you can now set up the utilities on your own computer. If you're fortunate enough to have root access on the systems at work, you'll also be prepared in advance to remove all traces of the *Sokoban* game from the accounting log files – just in case that evil corporate auditor really does show up in your department one day. [LXF](#)

About the author

Keith Gilbertson
(keithg@kellnet.com)
works as a programmer
analyst on the wireless
and Linux development
teams at the May
Company's Data Center
near Lake Erie.
On the lake, the fish
fear no penguins.

SOFTWARE INSTALLATION

Compiling software from source

Nick Veitch presents a beginners guide to the processes and potential pitfalls of compiling software.

One of the perennial problems that seems to crop up again and again in the magazine is people having difficulty installing software – usually software they have to compile themselves. In this tutorial, we'll be looking at what actually happens when you compile software, and explaining the steps as we go. A greater understanding of what is actually going on will hopefully give you a better chance of successfully installing software in the future.

Source files

Most software for Linux and other open source platforms, (and Unix systems in general) is distributed as source code, usually in a compressed archive commonly known as a tarball. This is just a collection of source code and data compressed into a single file for easy transportation.

As well as being put into the tar archive format, these files are often compressed with *gzip* or *bzip2* to save space and download times. The files will then commonly end in '.tgz' or '.tar.gz' for *gzipped* files and '.bz2' for *bzipped* ones.

Before installing any software, you will normally want to log in as the root user. Only root can install the software globally on the system so every user can access it. So before you go any further, it may be a good idea to open a terminal and use:

```
su -
```

and enter the root password. Now you can copy the tarball file from CD or download it from a website onto your hard drive. A good place to put files like this is in `/usr/src` directory. Okay, lets assume you want to compile *Gamazons* from the *LXF CD35C*.

```
cp /mnt/cdrom/Games/Gamazons/gamazons-0.77.tar.gz /usr/src
```

Now we can uncompress it using

```
tar xvzf gamazons-0.77.tgz
```

A lot of text will appear on the screen, showing the files as they are created. Now enter the gamazons directory and you are ready to begin compiling the code.

Compiling

What does compiling software actually do and why does it seem so problematic? A computer program is usually written in a 'language', like C. The computer itself doesn't understand this language. The CPU only understands a limited number of instructions, most of which are simply about moving data from one place to another, or comparing two values. What's more, this 'Instruction set' is usually unique to a particular processor. There are slight differences, for example, in the instructions understood by a standard 486 processor and a Pentium, and vast differences between an AMD Athlon and a PowerPC G3 processor for example. Writing in a language like C, which is then translated into the processor's code, means that the same software can be 'transported' between different processors easily. When running on the same OS, well written code will compile no matter what the processor actually is.

That's the main reason why a great deal of software is available as source code for Linux, because Linux itself can run on so many different processors. If you compiled code for a PowerPC chip and distributed it, nobody with a Zaurus PDA or an Itanium server would be able to run it! Distributing the source means that it can be compiled on many different system. It can also be made to work on other Linux systems with little alteration.

Using a compiler

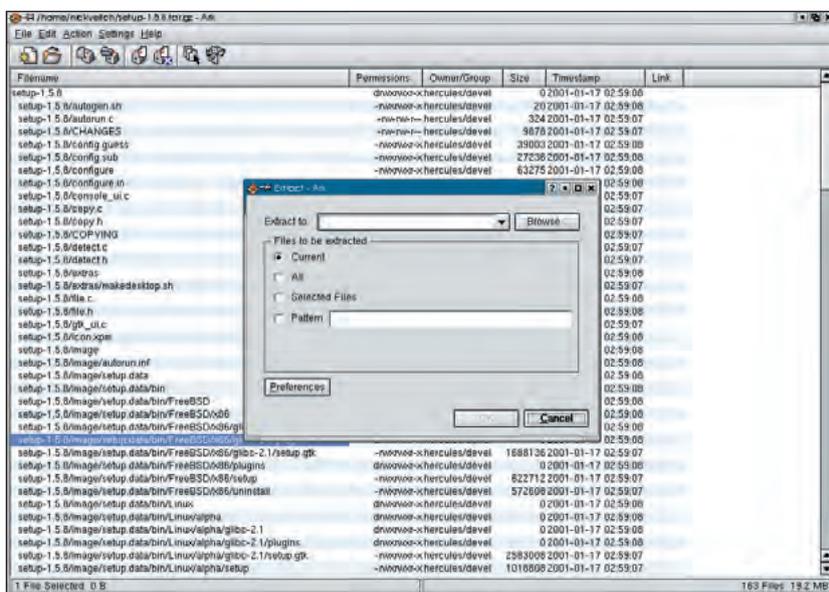
The most popular compiler for use with Linux is the excellent *GCC*. The *GCC* program simply takes the source code and compiles it into assembly language for the processor (well, it isn't simple, but that's a problem for the authors of the compiler, not you!).

Right. Consider a very simple program, like this minimal example:

```
int main(){
    printf("Hello, readers!\n");
}
```

It's a very simple C program which outputs a line of text when run, and is probably about the simplest C program you could

cover feature



If you find it easier, you might consider using a graphical program like *Ark* to extract files from a tarball archive.

write. In order to compile it, all we have to do is save this text in a file, and then run `GCC` supplying the filename as an argument:

```
gcc test.c
```

(Assuming we saved the text in a file called `test.c`. The program will compile and you will now have a binary, which by default is called `a.out`. If you want to give the output a different name, then you can specify this with the `-o` flag:

```
gcc test.c -o test1
```

Now you have a binary file, which you can execute on your computer, and it will work. However, if you take this binary and mail it to a friend, you may find that it doesn't work for them. If they are using a different processor, the code won't be understood and just won't work, which is why source distributions are often used.

Libraries

Very few pieces of software are like our example above. One of the great benefits of free software is that it is for sharing, and programmers share their code prolifically. The benefit to programmers is the re-usability of code. There is no point spending ages writing code that someone else has already written. For this reason many operating systems rely on libraries of code. You may have seen shared libraries on other platforms (for example, in Windows, all those `.DLL` files are libraries).

Libraries make code available through functions which the main program can call, just as if the library was included into the program. The example shown in Listing 1 (right) uses the `SDL` library (a graphics and sound library used extensively by games programmers) to create a window on the screen. You can see that the listing is only thirty lines long, but it opens a window on the screen, with a titlebar and gadgets for minimising and closing it. Clearly the code to do all that is not contained in the listing. It is actually included in the library, along with all sorts of other functions.

In order to make the functions available, the programmer just has to include the header file at the top of the code (the `#include <SDL/SDL.h>` line). But the actual core code of the library isn't actually included in the program – it is loaded from the library each time you run it. In order for this to work, the compiled program code has to 'link' to the library code. This means the compiler has two jobs to do – first it compiles object code from the source, then it links it and creates executable code:

```
gcc -c sdlcode.c -o sdlcode.o
```

```
gcc sdlcode.o -L/usr/lib/ -ISDL -o sdltest
```

The first line includes the options to compile the object code. The second line then links this code with the `SDL` library. If you don't specify the libraries to link to, you will get errors as the compiler won't be able to find the code for the functions you have used. The `-L` flag tells the compiler where to look for the library, though most libraries are included in the default library path and this can often be omitted. It is possible to combine these stages into one command, although the compiler actually completes the same two stage process:

```
gcc sdlcode.c -o sdltest -L/usr/lib -ISDL
```

will result in the same executable file. The library system works well. It allows programmers to re-use code, and it makes the actual source files smaller and more manageable. Imagine a simple test like the one in the listing which did have to include all the code for talking to the X server, opening the window and everything else!

Listing 1

A short SDL demonstration

```
#include <SDL/SDL.h>

main(int argc, char *argv[])
{
    int done=0;
    SDL_Surface *screen;

    if ( SDL_Init(SDL_INIT_AUDIO|SDL_INIT_VIDEO) < 0 ) {
        fprintf(stderr, "Unable to init SDL: %s\n",
        SDL_GetError());
        exit(1);
    }

    screen=SDL_SetVideoMode(640,480,8,SDL_HWSURFACE|SDL_DOUBLEBUF);
    if ( screen == NULL )
    {
        printf("Unable to set 640x480 video: %s\n",
        SDL_GetError());
        exit(1);
    }
    SDL_WM_SetCaption("This is our window",
    "TESTWINDOW");

    while(done == 0){
        SDL_Event event;
        while ( SDL_PollEvent(&event) )
        {
            if ( event.type == SDL_QUIT ) { done
            = 1; }
            break;
        }
    }
    SDL_Quit();
    return 1;
}
```

However, the downside to this is, if you don't have the library, you won't be able to compile the software. Libraries are the most common dependencies for software, which is why, if you take a look in your `/lib` and `/usr/lib` directories, you'll find an awful lot of them.

Library versions are also important. For the most part, libraries are backward compatible. That means that source written for a previous version of the library will usually work on the latest version too. The reverse is not the case though.

For programmers, it is often desirable to use the latest versions of libraries – usually because useful new features have been added, or irritating bugs have been removed. This may mean that although you have the right library to compile the software, it may not be the right version.

For the people who actually compile the code, this can be a nightmare. It's also a bit of a bind for the programmer, not the least of which because if they are creating large projects using many libraries, the compile command gets very long and



TutorialCompiling

◀◀ complicated. Fortunately there is a helping hand for everyone in the form of *Autoconf* and *Automake*.

Configure

These tools can be used to build a configure script for a software source distribution. This script can test for the presence of particular features and libraries, accept particular options which can be used to control features of the software, and create a Makefile which can be used to build the executable.

You have no doubt come across the configure script before when trying to compile software. The main function of this script is usually to test what sort of system it is being run on, whether the compiler supports various features and what libraries are installed. Usually, all that is required is to run the configure script itself:

```
./configure
```

Nine times out of ten this will probably work, and when it doesn't, it's usually quite obvious where the problem lies from the resulting error message. It is nevertheless important to read any documentation that came with the source code. Particularly software which is intended to be used on many different platforms (even ones which aren't Linux), there may be vital options, environmental variables or other things that must be set.

Try this is your gamazons directory that we uncompressed earlier. This game uses common GNOME libraries, so if you get any errors regarding these, you may want to update your GNOME packages for your distribution. Assuming this went correctly, all you have to do now is make the file:

```
make
```

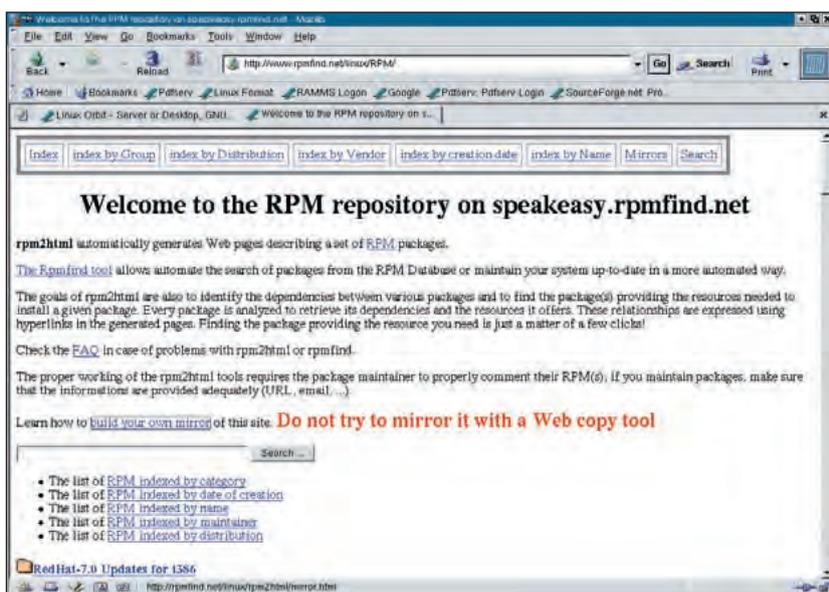
and then install it:

```
make install
```

This last step copies the executable to a general directory, in this case `/usr/local/bin`. If you want to install the software somewhere else, you can usually pass such information to the configure script.

When things go wrong

Well, that's how the compilation process is supposed to work – but what can you do when it doesn't? Well, the first thing



If you can't find particular libraries on your system or your LXF CDs, a search engine like www.rpmfind.net can be useful.



Loki's *Setup* installer worked very well, though it would be too much to ask for every release of free software to go to the lengths of building a dedicated installer – which is why learning to compile software is so useful.

to do, if you haven't already done so, is read the documentation. Most source distributions will come with a README and an INSTALL file. These often contain crucial information and useful options. There may be special configure options you need to use, or different things to do depending on your system. You should certainly read these files first if something does go wrong.

The most common failure is actually at the configure stage, when the configure script detects an out of date or missing library. Most common libraries will have been included with your distribution. If the library is simply out of date, you can usually update it from the update source for your distro. For example, Red Hat maintain a pages of updated RPM packages for most of their distros. Mandrake and SuSE both have online update tools. You will also usually find source and binary packages for common libraries on the *Linux Format* CDs and DVD in the Essentials folder.

Sometimes the configure script can be fooled if the library is present, but installed in an odd location (and therefore not in the list of paths that the library loading tool checks). Most distributions include the *locate* command or other tools which can be used to find files. If you are certain that you do have the correct version of the library, check with, for example:

```
locate libqt.so
```

which will return a path or list of paths where matching files can be found. Then you can take a look through the `configure.log` to find out how the configure script failed and if it was looking in the right place.

The configure script may well give you the option to specify exact paths for particular libraries, and other options, for example.

```
./configure -qt-libraries=/usr/lib/qt3/lib
```

Note that although sometimes you may have the libraries themselves, to compile software which makes use of them, you also usually need the development version, which will include header files the library was built with.

Summing up

Hopefully you will have found this short tutorial useful. It has given you some background on how to compile software and how the actual system works. By giving you some background on this, you should be more able to solve any problems you come across. Trying to get some software to compile can be frustrating, but the vast majority of the source code you encounter will work fine provided you have the necessary libraries and the patience to read the instructions. [LXF](#)

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 modern 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.



Nick Veitch is the editor of the magazine, and answers your easy questions!

Or indeed anything to do with *Grub*, *LILO*, *netatalk*, *vi*...



DVD backup

Q DVD-RW units are coming down in price to such an extent that they represent a preferable way of backing up a system. I am looking to spend between £250 and £300 on such a device and would prefer it to be an internal unit.

Could you tell me which of the DVD-RW units are compatible with Linux and what software/drivers would be required?

Any info on configuring such a device would be appreciated.

Mike Bennett

A You're most likely to have success with an internal ATAPI/IDE device, as most external DVD-RW devices are IEEE1394, otherwise known as firewire, which is somewhat questionable under Linux. With an ATAPI device, you can set it up to use the *ide-scsi* module, just like a CD burner, but you need to do things significantly differently in order to be able to write data to the device, as DVDs don't handle sessions like CDs do.

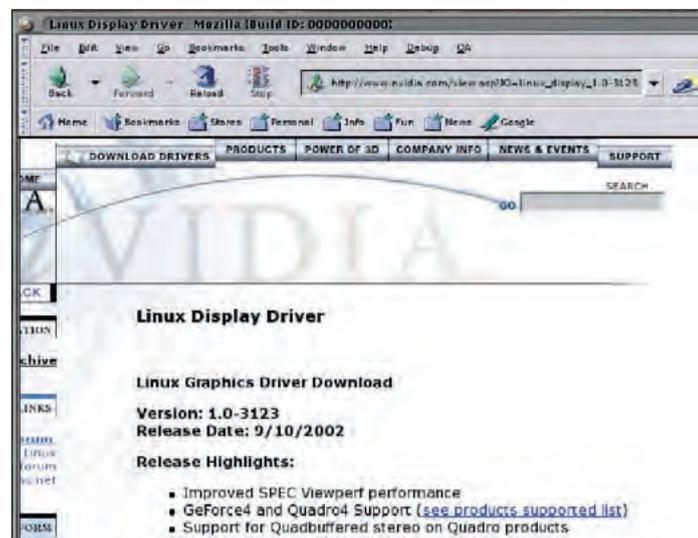
You need to apply a patch to your kernel, as well as obtain some tools in order to build a UDF file system on the DVD. Once you have a file system on the DVD, you can simply mount it and copy files to it. An item of DVD-RW can be written around a thousand times, so there are some things you have to do in order to stop it being written by the system when you don't want it to be.

Lots of info, and links to all the packages you need, are at: <http://fy.chalmers.se/~appro/linux/DVD+RW/>

OpenGL libraries

Q I am very new to Linux and have recently installed Red Hat Linux 7.3 which is running well, printer, video card etc.

I was keen to try *Flight Gear* from the Oct 2002 LXF DVD-ROM.



nVIDIA have binary X drivers for their popular GeForce range of video cards, including GeForce4.

After mounting the DVD drive, I was able to read the directories, and running the command:

```
sh Flightgear.sh
```

I received the error message:

```
./fgfs: error while loading shared libraries: libglut.so.3: cannot open shared object file: No such file or directory
```

Can you please advise me whether this file or directory is likely to be in the Red Hat distro, or is it on the DVD-ROM, and how I might locate and utilise it to enable me to run *FlightGear*.

Ron Wilkins

A The *FlightGear* simulation program requires the OpenGL libraries to be installed on the system, including *libglut*. This can be installed by obtaining the *Mesa* package from either your installation CD, a Red Hat mirror server, or from rpmfind.net. Ensure that you only install the *Mesa* package for the RedHat distribution, as if you try to install a package for Mandrake or SuSE, things are going to get awfully confused and it's not unlikely that lots of dependencies will break.

This will allow *FlightGear* to be run, although you may find that there are other libraries which need to be installed. You can quickly find out which package provides which libraries by going to rpmfind.net and putting in the library name, such as *libglut*, and searching for Red Hat 7.3 packages which contain files with that name.

Graphics drivers

Q I have a problem with a graphics card of a similar nature that was reported in your April 2002 edition.

My son bought a G4MX420D-T graphics card, from the Taiwanese company Micro-Star International. The card worked brilliantly in Windows 98 SE but when I tried to boot into Linux it packed a sad, as the GUI logon screen would not appear.

The distro I am using is Mandrake 8.2, and my computer is a Compaq Deskpro, SB series with a Celeron 1.1MHz CPU and 384MB of RAM installed. The hard drive has a capacity of 20GB and is partitioned with Windows

98 SE in the primary partition.

I have checked the XFree86 site and there isn't any mention of Micro-Star, nor any mention of nVIDIA GeForce4, which is supposed to be the standard of the above-mentioned graphics card. Micro-Star's packaging claims the card to be compatible with Linux but if this is all rubbish, under New Zealand consumer laws, any false claims made by a vendor gives the purchaser the right to return the item and receive a refund.

Have you any suggestions (including sending the graphics card back to the retailer) to get round this problem?

Des Eyre

A The MSI video card is a very popular GeForce4 card, so you will be able to use it with nVIDIA's binary driver distribution. You can find this at www.nvidia.com/view.asp?IO=linux_display_1.0-3123, although you may want to go in from www.nvidia.com to ensure that you get the latest release. There are RPMs available, along with the kernel module for Mandrake 8.2, so you should not need to compile anything by hand.

```
david@macha:~ (pts/3)
naeve:~# free
              total        used        free       shared    buffers     cached
Mem:          1551592      1491708      59884           0        174742      911548
-/+ buffers/cache:  485376      1146216
Swap:         6291408      935252      5356156

naeve:~# swapon -s
Filename                                Type              Size      Used      Priority
/dev/naeve/swap                          partition         2097136    0         -1
/dev/naeve/swap1                          partition         2097136    0         -2
/dev/naeve/swap2                          partition         2097136    0         -3
naeve:~#
```

There are a number of utilities which can be used to check your swap space, even if the kernel isn't using it.

Swap

Q I'm running SuSE 8.0 on a PIII but despite a listing in fstab and the fact that I specifically made a swap partition I don't believe that the swap memory is being used, causing some problems. Luckily, I've a reasonable amount of RAM 512MB@133MHz. I monitor resource usage and watch as all my RAM is used up but no SWAP.

So, how do I make sure that my swap partition is being used or capable of being used. How do I initialise it without starting from scratch and reinstalling – only ever

had this problem with SuSE.

Any advice gratefully received.

Daithidh MacEochaidh

A You can check that your swap space is being used by running:

```
swapon -s
```

as root. Chances are, your swap partition is being used by the system, but the kernel doesn't see any point in swapping things to disk.

The vast majority of your memory is probably being filed for file caching and other buffers, rather than memory which has been allocated by processes. The kernel will dump things out of the buffers before resorting to

swapping to disk, as this is obviously much more efficient. You can check that swap is working by running a big application, such as Mozilla or The GIMP, and leaving the system for a number of days. Generally, memory pages will be swapped out to disk when the application which is running has not accessed those pages for a long period of time, so after a number of days, something which is just sitting there will probably be swapped to disk.

Alternatively, you can attempt to consume a lot of memory by mounting /tmp as a tmpfs file system and using dd to create a large file in it, which will use up a lot of your available memory. You could then compile a kernel, or build another large package, as compiling code requires a great deal of memory and the system will likely swap parts of the large file you created out to disk to make room for the memory required by the compiler.

You can always monitor your memory usage with the free command and you can quickly check how much swap space is available, what is used and what parts of your main system memory are being used for buffers and file system caching.

	total	
used	free	shared
buffers	cached	
Mem:	1551592	
1490980	60612	0
185860	949224	
-/+ buffers/cache:	355896	
1195696		
Swap:	6291408	
1172256	5119152	

Library versions

Q Firstly I am a complete beginner to Linux, however I tried to install the KDevelop IDE onto my Mandrake 8.2 system but got the error message kdevelop <=>S 2.0.2-4mdk conflicts with libpng3-1.2.1-6mdk

I am then asked to either force or quit, being a chicken I opt for quit. Can I force the installation without any harm?

Gary Steel

A When there is a conflict, it depends which file they are arguing over. You might want to check to make sure that you have the latest KDevelop and libpng3 releases, although you have the Mandrake releases of each package so one would hope that there would be no conflicts with packages built for

A QUICK REFERENCE TO: DVDs

Watching movies on a computer can be great fun. Whether it's having your favourite movie on your laptop when taking a trip or using the company's LCD projector to pretend that you are at the cinema watching the latest release. Of course, playing DVDs is not without problems, particularly down to the use of CSS, an encryption method used by studios to stop people copying DVDs digitally. However, this encryption is rather poor, by most standards, and it didn't take a young Norwegian very long to create a crack for it. While this did result in him being prosecuted, we now have an Open Source method of decrypting DVDs on Linux systems, which is usually distributed in the form of libdecss.

There are a number of players for Linux which can handle DVDs and DeCSS, the most popular of which are Xine (<http://xine.sf.net/>)

and mplayer (www.mplayerhq.hu). Both have their pros and cons, but generally, they both work very well and each can take advantage of various DVD capabilities which the other lacks. A particularly problematic DVD feature is interactive menus and Xine's dvdnav plug-in (<http://dvd.sf.net/>) is a good choice, although it does have a number of minor issues when dealing with some DVDs. Unfortunately, by getting DVD menus to work, we lose out on some speed improvements which the standard d4d player provides. dvdnav has been included into the main Xine distribution over the last few months, although it does not natively support DeCSS due to legal issues surrounding the distribution of the code within the US. However, individual users are able to install the DeCSS libraries on their own, which Xine will make use of.

Many people find mplayer to be much faster than Xine, not to mention

simpler to install, as it doesn't have a whole array of dependencies. Both mplayer and Xine can play other formats, such as AVIs and DivX files, so we don't need to have half a dozen different apps to play various formats.

It is also possible to copy DVDs and create DivX files, which can be stored locally and played without the DVD being present. A DVD in DivX format may take up around 1GB, although this depends upon compression levels and the resolution used, so we could copy a selection of DVDs and store them all on a large hard disk for quick access. The most popular package to do this is transcode (www.theorie.physik.uni-goettingen.de/~ostreich/transcode), which can process a large number of video and audio formats into other formats. A DVD can be converted to a format which can be played on almost anything, even a low-end system without fancy software decoders.

FREQUENTLY ASKED QUESTIONS: BURNING CDS

FAQ What software can be used to burn CDs in Linux?

There is only really one software package which can be used to create CDs in Linux, which is *cdrecord* (www.fokus.gmd.de/research/cc/gclone/employees/joerg.schilling/private/cdrecord.html). This is a very capable command line tool, which can be used to burn ISOs to CDs, as well as separate tracks, such as for an audio CD, and supports all of the major CD standards.

FAQ Command lines scare me! Is there one with things I can click on?

cdrecord has a number of front-ends which can be used with KDE, GNOME and other desktop environments available under Linux. Many of these have other features, including ISO creation and ripping of audio CDs built-in.

A list of many of the *cdrecord* front-ends is available at <http://sites.inka.de/~W1752/cdrecord/frontend.en.html>. Popular front-ends are *KEasyCD* and *gtoaster*, both of which have a wide range of features and have been well tested by the user base. Of course, it's a good idea to download a selection of

them and decide for yourself which is more ideal for your needs.

FAQ Why can't cdrecord find my CD drive?

cdrecord only supports SCSI interfaces, so if you have a ATAPI/IDE device, you will need to use the *ide-scsi* kernel module to provide emulation of a SCSI device. It is important that the IDE subsystem does not use the CD drive, otherwise *ide-scsi* can not claim it and use it as a SCSI device. This can be done by adding **ignore=hdd** to the kernel command line, either manually, or with an append line in the *lilo.conf* file. Naturally, one should change **hdd** to the IDE device which is for the CD-RW drive.

If *ide-cd* is compiled as a separate module, we need to ensure that we setup */etc/modules.conf* to accept 'ignore=hdd' as an option:

```
options ide-cd ignore=hdd
```

Once this is done, we can load the *ide-scsi* module and it will locate our CD drive, which will be available as */dev/sr0*, so the link to */dev/cdrom* can be modified to make use of this new device.

cdrecord requires a proper SCSI reference, including host adaptor ID, channel and LUN. Usually, this will be 0,0,0 on a system without a SCSI adaptor, but we can use a *cdrecord* option to search for SCSI devices:



cdrecord -scanbus

This will list all SCSI information, so we can quickly find the CD-RW drive in the list. We can then run *cdrecord* with this as the **dev=** option.

FAQ Can I load ide-scsi automatically when I use cdrecord?

Depending upon the distribution, it may be possible to load *ide-scsi* when we boot the system up. If it's important that *ide-scsi* is available from the outset, it's probably worth just compiling it into the kernel.

If we're using a system with *devfs*, we can simply have it load *ide-scsi* when a device within */dev/sr* is accessed. This can be done with an alias in *modules.conf*:

```
alias /dev/sr ide-scsi
```

For non-*devfs* systems, we need to alias the block major which is used for */dev/sr** devices to *ide-scsi*:

```
alias block-major-11 ide-scsi
```

In either case, we also need to load other support modules:

```
pre-install sg modprobe ide-scsi
```

```
pre-install sr_mod modprobe ide-scsi
```

FAQ How can I use cdrecord to burn an ISO?

All you need to do is use the ISO file as an option on the command line for *cdrecord*:

```
# cdrecord dev=0,0,0 file.iso
```

FAQ I need to create an ISO of a directory. How do I do that, so I can burn it to a CD?

◀ the same distribution – so much for a simple life.

You could try to force the install, but make sure that you have a copy of the *libpng* RPM handy just in case things are not happy with whatever *kdevelop* installs. If you don't want to force it, your best bet is to contact whoever built the *KDevelop* package and ask them why you see the conflicts. You can find who built it by using the `rpm -qip <filename>` command.

Firewall boot

Q I've been receiving the mag for over a year now and have found it to be very helpful.

My interest is with Linux use on older, so called out-of-date hardware, and the very useful tiny distros you've made me aware of.

I've now met with a problem I hope you can help me with. I am intending to use a Compaq prolinea 466 as a firewall. Having got the PF10 set-up from the HP/Compaq website, I find that the CDROM is unbootable. So, no problem I thought, load *tomsrtbt* and do a *Lilo* boot: `zImage hda hdc=cdrrom` command based on Hoyt Duff's article in *LXF23* p68. This resulted in the message `"/etc/lilo.conf: No such file or directory"`

Of course *tomsrtbt* does not have such a file, so create one, but what should be in it? The man pages I've looked at have not helped me, just created more confusion. Can you suggest what should be in this file for the *Lilo* command to work?
Malcolm

A Tom's Root and Boot is not a distro that you can install, so does not have *Lilo* available. If you are unable to boot from CD, most distributions have boot images on the CD which can be written to a floppy using a DOS utility such as *RaWrite*, or if you have another Linux box available, using *dd*. You can then boot from the floppy, point it to the CD and install it as usual. If you only want to use it as a firewall, you may want to look at certain distributions which are designed for use as a dedicated firewall system, such as *Smoothwall* or *IPCop*. Both of these have very good documentation and as many people will run them on older hardware, they can both be installed from CD after booting from a floppy disc. You could also install a distribution such as

Debian or Red Hat, using a boot floppy, and then set the firewall up manually, although this will depend upon your own experience and tastes.

Zippping slowly

Q I have recently obtained the new USB Iomega 750 zip drive to use on my P733 PC on which I am running SUSE 8.0.

After plugging everything in, I put a 750MB disk into the device and entered as root:

```
mount -t vfat /dev/sda4 /zip750
```

The device was ready to use.

Unfortunately the writing speed to the device appears very slow (over 7 minutes for a 30MB file).

Is there anything that I can do to speed this up? I purchased the device in order I could easily backup

A utility called *mkisofs* is used to produce a ISO file from a directory source. We can build an ISO with a number of options, including Joliet or Rockridge extensions permitting for long filenames, as ISO9660 only permits 31 character filenames, although *mkisofs* uses 8.3 filenames as default for compatibility with MS-DOS.

We can create an ISO with:

```
$ mkisofs -o file.iso dir
```

If the system is fast enough, we can create an ISO and pipe it straight into *cdrecord*;

```
# mkisofs dir | cdrecord
dev=0,0,0 -
```

FAQ Can I make audio CDs with *cdrecord* using MP3s?

Using the *mpg123* utility, it's easy to create a file which can be burned to a CD. We need to do this in two stages. First we have to burn the music to the CD, then we have to 'fix' the CD when we're done.

```
mpg123 -cdr -s file1.mp3 |
cdrecord -nofix dev=0,0,0 -
mpg123 -cdr -s file2.mp3 |
cdrecord -nofix dev=0,0,0 -
mpg123 -cdr -s file3.mp3 |
cdrecord -nofix dev=0,0,0 -
mpg123 -cdr -s file4.mp3 |
cdrecord -nofix dev=0,0,0 -
cdrecord -fix dev=0,0,0
```

It is very straight forward to produce a simple shell script which can burn a directory of MP3s to a CD.

large files quickly. I appreciate that I am probably not running USB 2.0 for which it is intended and that Linux is not officially supported but other Iomega zip drives that I have used worked very well under Linux.

This is the first time that I have plugged anything into the USB port and thus I hope that I am not doing anything silly here which is causing the slow write speed.

As a comparison, I can write to my internal 250MB Zip at nearly 1MB per second and I can write to my CD-RW nearly as quickly as this when I use the **speed=4** option with *cdrecord*:

```
cdrecord -v -eject -force -multi -pad
-data speed=4 dev=0,0,0
/tmp/file0000.img
```

Michael Warby

Regular USB can perform at around 15Mbit/sec, so you should be able to transfer quicker than 73kb/sec. This likely down to the kernel module you are using for USB, so you may wish to switch to *usb-uhci.o*, rather than using *uhci.o*, and see if that performs any better. You may also want to check *dmesg* to ensure that the USB controller is not going crazy with errors, or something else is occurring which could slow things down.

If that does not help, your next port of call is hardware, so check the cable and ensure that you use the cable that came with the Zip drive, otherwise it may slow itself down in order for data to be transferred reliably.

Two mice

I have a Dell Latitude C800 with integrated Glidepoint mouse (pad and nipple), using Mandrake 9.

Can I set up X so that if I plug an external mouse in it will switch to it, and back when I unplug it?

At the moment when I plug the external mouse in the Mandrake hardware thingy pops up and replaces the Glidepoint mouse settings in XF86Config-4 with external mouse settings. I then have to run *mousedrake* to get the integrated mouse working again.

Can I put two "mouse" entries in XF86Config-4? From the LXF forums.

The pointing devices on most laptops are PS/2 devices, so when you plug a mouse into the PS/2 port on the laptop, the laptop will handle switching from the internal device to the external mouse. There is usually a BIOS option to determine if the internal device will be disabled, or if they will both work together.

If the external mouse is USB, you will need to have the *mousedev* and *input* modules available or compiled into the kernel, then set X up to use */dev/input/misc* as an additional mouse in XF86Config:

```
Section "InputDevice"
Identifier "Generic Mouse"
Driver "mouse"
Option "SendCoreEvents"
"true"
Option "Device"
"/dev/input/mice"
Option "Protocol" "ImPS/2"
Option "Emulate3Buttons"
"true"
```

```
david@macha:~ (pts/23)
Option "CorePointer"
Option "Device" "/dev/misc/psaux"
Option "Protocol" "PS/2"
Option "Emulate3Buttons" "true"
Option "ZAxisMapping" "4 5"
EndSection

Section "InputDevice"
Identifier "Generic Mouse"
Driver "mouse"
Option "SendCoreEvents" "true"
Option "Device" "/dev/input/mice"
Option "Protocol" "ImPS/2"
Option "Emulate3Buttons" "true"
Option "ZAxisMapping" "4 5"
EndSection

Section "Device"
Identifier "Generic Video Card"
Driver "i810"
VideoRam 8192
Option "UseFBDev" "true"
EndSection
```

Adding extra mice to a system running XFree86 4.x is easy, although the device must exist when you start X.

```
Option "ZAxisMapping" "4 5"
EndSection
```

You will need to have both *mousedev* and *input* loaded before XFree86 starts up, otherwise it will die when it tries to open */dev/input/mice*. Once it works, you can just plug in a USB mouse and X will automatically allow it to be used in addition to the internal mouse on the laptop.

Promised IDE

On a 2 hard-disk drive PC with Windows XP Pro and a Promise "Ultra133 TX2"

ATA controller card (www.promise.com/product/subsys_detail_eng.asp?pid=87&fid=3) the Debian Installation System 3.0.23 (built on 15/05/02) that uses kernel 2.4.18-bf2.4 from the cover DVD of the November edition of LXF gives the error message:

"No hard disk drives were detected" after the:

```
"Configure the Keyboard"
selection is made.
```

When an attempt to access the floppy-disk drive for the ATA controller's device drivers is made, there is an error message that states that the floppy-disk drive cannot be mounted. (The floppy-disk was formatted for an IBM compatible PC.)

The boot argument "floppy=thinkpad" produces the error message: "Could not find kernel image: floppy=t.him"

Another boot argument "bootfloppy0" produces the error message: "VFS: Insert root floppy and press ENTER request_module[block-major=2]:

```
Root fs not mounted
VFS: Cannot open root device
"fd0" on 02:00
```

Please append a correct "root=" boot option

Kernel panic: VFS: Unable to mount root fs on 02:00" and then the system hangs.

The kernel image "idepci" is not available as a boot argument. Is there another way to install Linux apart from obtaining an official version of the program? (This is my first experience of Linux.)

Daniel P. Wheeler

The kernel module for this Promise IDE adaptor is distributed for Debian as part of the *drivers.bin* files, although it seems you built this floppy disk incorrectly. You can't simply copy the file to the floppy disk, as it must be written using a utility such as *RaWrite* in DOS or *dd* in Linux. The *drivers.bin* file contains the file system used by Debian to store its distribution's contents, so it expects to be able to mount the floppy disk with that file system.

You will want to read the install guide for Debian, which can be found at www.debian.org/releases/stable/installmanual. As you will be booting from this adaptor, you will need to ensure that the kernel Debian boots off either has support for the adaptor compiled into it, or can load the module via an *initrd* at boot time so that it can be mounted by the kernel for use as a *rootfs*.

File copying

Can anyone tell me a way of moving files between networked computers. I have a laptop and desktop hooked up directly to one another and can

Answers

« ping etc. I have been using NFS to mount part of one systems filesystem on the other machine but this is a pain when working with removable media as it won't let me unmount the media without rebooting. It probably would if I knew what it was that needed to be switched off but I don't.

What I am looking for is a simple utility like "send file" in Pocket PCs. Just tell one machine to send and the other to receive and wey-hey the file is transferred.

Any help is much appreciated but I am new to networking so need something simple.

From the LXF forums.

A For copying files back and forth between systems, the *scp* utility from *OpenSSH* is often the best choice. This requires that the machine you're copying files to has the *SSH* daemon installed, so that the client can connect to it before copying the files over the secure *SSH* connection. Each *scp* connection will require authentication, so it will ask for the password on the remote machine in order to authorise the *SSH* connection. This is really annoying when copying a lot of files, so using *SSH* keys is a much better choice, as *SSH* will authenticate without you having to do anything, allowing you to *scp* files back and forth quickly. You can specify any destination location to write the file to, assuming the user you *SSH* to on the remote system can write to the location.

For large file transfers, *rsync* is also a good choice, as it allows copies to be resumed, so if you have a big directory with a load of files, you can stop it and start it whenever you like. *rsync* is generally run over *SSH*, so the

need for key-based authentication with *SSH* still exists.

Shared memory

Q I am using RH 8.0, I am using the command *df* to see my free space. That when I notice a directory called */dev/shm*. I checked *fstab* and this is the info I got

```
none /dev/shm tmpfs defaults 0 0
```

I want to know what this is, is it my swap partition?

From the LXF forums.

A SHM is for Shared Memory, which is memory allocations shared between processes on the system. *tmpfs* is a filesystem which grows when required and is allocated out of system memory, rather than living on a physical disk. A process can create a file in */dev/shm* and another can read it, assuming the file access properties are appropriate for such an operation, allowing data to be shared between devices. This can be performed with a regular filesystem, but it will be much slower as it will need to be written to disk, then read back from disk for the other process.

tmpfs is also often used for */tmp* and */var/tmp*, as it's much quicker for temporary file storage than a */tmp* mounted from a disk partition. *tmpfs* also used to be known as *shmfs*, as it was originally developed for use with */dev/shm*. You can find more information on *tmpfs* by checking *Documentation/filesystems/tmpfs.txt* within the Linux kernel source tree.

Hardware time

Q Odd one, this. I use *ntpdate* to set my clock when I connect to the Internet. Last

```

david@macha:~ (pts/22)
tmpfs is a file system which keeps all files in virtual memory.

Everything in tmpfs is temporary in the sense that no files will be
created on your hard drive. If you unmount a tmpfs instance,
everything stored therein is lost.

tmpfs puts everything into the kernel internal caches and grows and
shrinks to accommodate the files it contains and is able to swap
unneeded pages out to swap space. It has maximum size limits which can
be adjusted on the fly via 'mount -o remount ...'

If you compare it to ramfs (which was the template to create tmpfs)
you gain swapping and limit checking. Another similar thing is the RAM
disk (/dev/ram*), which simulates a fixed size hard disk in physical
RAM, where you have to create an ordinary filesystem on top. Ramdisks
cannot swap and you do not have the possibility to resize them.

Since tmpfs lives completely in the page cache and on swap, all tmpfs
pages currently in memory will show up as cached. It will not show up
as shared or something like that. Further on you can check the actual
RAM+swap use of a tmpfs instance with df(1) and du(1).

</Documentation/filesystems/tmpfs.txt [creadoniy] 182L, 4181C 1, 1 Top

```

tmpfs uses system memory to store files, similar to a RAM disk, making it great for */tmp*.

weekend, I dialed up to the Internet and *ntpdate* pushed my system clock back an hour to GMT, as expected. However, whenever I reboot my machine, I find the clock has reset to BST (GMT + 1 hour). This is annoying.

How do I stop this?

From the LXF forums.

A Each system has two clocks. One is a software clock, maintained by the kernel running on the system, the other is a clock maintained by the BIOS, known as the hardware clock.

ntpdate updates the software clock, so when you reboot the system, it won't be saved.

The utility *hwclock* can be used to write the current software time to the hardware clock with the **--systemd** switch. Usually, this is performed by distributions when you shut the system down.

Another consideration is the current state of your hardware clock. Generally, the hardware clock should be on UTC, so it won't change when we change our clocks for daylight savings time or back to GMT in the autumn. If your operating system has its time zone set to GMT, but your hardware clock is running on BST, then things will be one hour ahead.

It's worth remembering that if you have your time zones set up correctly on your system, *ntpdate* won't actually change your date when we switch from GMT to BST, or back again. *ntpdate* will always deal with time zones, so you can happily update your GMT system with a NTP server which is running on PST. If you set your distribution's time zone to GMT or BST, it will not automatically

change, but if you set it to 'GB', or if there is an option to select a location, then the system will switch from BST to GMT automatically. If you don't do this, then you will need to manually change your system's time zone with the *tzconfig* utility when the clocks change, but you will not need to change either the BIOS clock or the software clock, as the change of time zone will account for the one hour change. [LXF](#)

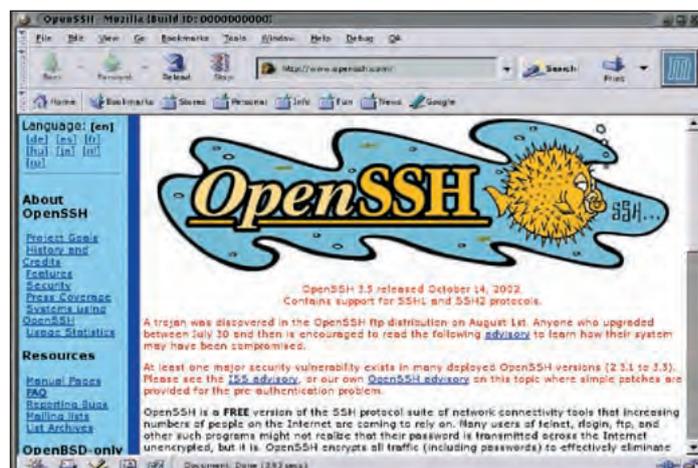
Submission advice

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



scp, distributed as part of *OpenSSH*, is a great way to copy files between two systems.

missed one?

LINUX FORMAT BACK ISSUES

Every month *Linux Format* brings you the best tutorials, the essential reviews and the latest news. But if you've missed out on a must-read feature or a vital programme from our expertly compiled CDs and DVDs, order your back issue NOW! And remember, you need never miss an issue of your favourite Linux mag, if you subscribe to *Linux Format* (see overleaf for more details).



December 2002

Product code:
LXFB0034(cd)
LXFD0034(dvd)

DVD HIGHLIGHTS:
KDE 3.1 beta2, Scribus, Smoothwall2, Movix, GNUPrivacyGuard, GCompris, Drip

MAGAZINE FEATURING:
Distro reviews of SuSE 8.1, Red Hat 8.0, Lycoris and a mini-distro RoundUp, scanning, PHP acceleration, Linux Expo UK 2002

CDs HIGHLIGHTS:
Unreal Tournament 2K3 (Demo), DemoLinux, Cardfile, RUTE, OOoDictionaries, Mozilla binary packages



November 2002

Product code:
LXFB0033(cd)
LXFD0033(dvd)

DVD HIGHLIGHTS:
Debian 3.0, mFighter, OpenOffice.org (bugfix release), DVDrip, Cyrus IMAP Server

MAGAZINE FEATURING:
Sun's move into the Linux server market – with LX50 review, The Liberty Alliance, Systems programming, using OpenOffice.org, Homepage

CDs HIGHLIGHTS:
Kylix 3, GNUCash, BXPPro, KDevelop, Opera, Vega Strike, Parted, AnjutaIDE, GTransferManager



October 2002

Product code:
LXFB0032(cd)
LXFD0032(dvd)

DVD HIGHLIGHTS:
Knoppix, Drip, Squeak, extra FlightGear maps, Ogg Vorbis 1.0, Knoppix

MAGAZINE FEATURING:
Building better databases, 'Trusted Computing' – beware Palladium, USB 2.0, firewall roundup, Amiga emulation, Gentoo review

CDs HIGHLIGHTS:
FlightGear (runs from disc), Aglaophone, UAE, Clam Antivirus, Perl 5.8, Quanta Plus, Netclipboard, Mah-Jong, HTML-Mason, WebSuck, Epsutil



September 2002

Product code:
LXFB0031(cd)
LXFD0031(dvd)

DVD HIGHLIGHTS:
Slackware 8.1, Cinelerra, Ogle & Zine, Gnome2, MultiCD, Phobia III, Grip, Zinf, TkVoice

MAGAZINE FEATURING:
Linux goes to Hollywood, HTML Editors roundup, Internet security special, Ruby scripting language

CDs HIGHLIGHTS:
Slackware 8.1, Fluxbox, Kallers, Torcs, SaveMyModem



August 2002

Product code:
LXFB0030(cd)
LXFD0030(dvd)

DVD HIGHLIGHTS:
Intel's C++ & Fortean compilers, CPAN, Gnome2, Gentool Linux, Boson, Mozilla, Normalize, EarCandy

MAGAZINE FEATURING:
Ultimate Office: Every current office solution on test, plus Quantum computing, Red Hat 7.3, PHP tips, Eden motherboard test

CDs HIGHLIGHTS:
Highlights: Intel's C++ & Fortean compilers, GNOME2, Transcode, Album Cover Grabber, Office Suites



July 2002

Product code:
LXFB0029(cd)
LXFD0029(dvd)

DVD HIGHLIGHTS:
Highlights: OpenOffice.org 1.0, Mozilla 1.0rc3, Netscape 7.0, Opera 6.0, Mac on Linux

MAGAZINE FEATURING:
Customise your kernel, WineX latest, Inside IPv6, Astronomy applications roundup

CDs HIGHLIGHTS:
Highlights: OpenOffice.org 1.0, Evolution, Omnis Studio 3.01, Clam AntiVirus, Python 2.21

Other back issues also available, please call for further information, giving the issue number and/or date required

To order, call the back issues hotline on **0870 4448645**

or **+44 1458 271178** for overseas orders

Remember to quote the issue code shown above and have your credit card/switch card details ready.

Prices (inc post and packing): CD – UK£5.99, US & Eur £6.99, rest of world £7.99
DVD – UK £6.99, US&Eur £7.99, rest of world £8.99.

LINUX

www.linuxformat.co.uk **FORMAT**

Coverdisc



Neil Bothwick is your guide through the wonders of this month's jam-packed *Linux Format* CD, as we look up from the desktop and reach for the stars...

Not only do we have two full CDs of Mandrake, we also have a packed CD containing the normal selection of new and interesting software, along with files associated with articles in this issue of the magazine.

Internet/SpamX

One measure of the ever-increasing volume of spam, is the increasing number of anti-spam solutions that appear. Most of these are intended to be run from a mail transfer agent, like *Postfix* or *Exim*, or a mail delivery agent, like *Procmail*. That's fine if you have your own server, or your ISP allows you to upload *Procmail* recipes, but what if you simply download mail from your POP3 mailbox to your mail program? Short of setting up your own server, these solutions are generally not applicable. *SpamX* is an exception to this generalisation. It works by logging into your POP3 mailbox, checking for and deleting spam and then pausing for a specified time before looping around and doing it all over again.

SpamX's filtering capabilities are nowhere near as sophisticated as those of *SpamAssassin* or some of the pre-defined *Procmail* recipes, but neither is it in any way difficult to set

```
SpamX: Working for account [Freemail], looping every [10] minutes...
SpamX: Connecting to freemail.gr port 118
SpamX: Logging in...
SpamX: 7 messages on server, total [2256] octets
SpamX: Checking mail...
SpamX: Logging out...
SpamX: Working for account [Popper], looping every [10] minutes...
SpamX: Connecting to popper.forthnet.gr port 118
SpamX: Logging in...
SpamX: 15 messages on server, total [5428] octets
SpamX: Checking mail...
SpamX: Logging out...
SpamX: Working for account [Work], looping every [10] minutes...
SpamX: Connecting to pop-forthnet.forthnet.gr port 118
SpamX: Logging in...
SpamX: 11 messages on server, total [4084] octets
SpamX: Checking mail...
SpamX: Message 11 from [@tremendousbuys.com], is in spammers list, deleting
SpamX: Bouncing back to [newsletter@tremendousbuys.com]... Done.
SpamX: Logging out...
tevas@sylvester evas:~$
```

Say goodbye to junk email with *SpamX*.

up. It won't catch all of your spam, but any reduction in this infernal blight on the email system is welcome.

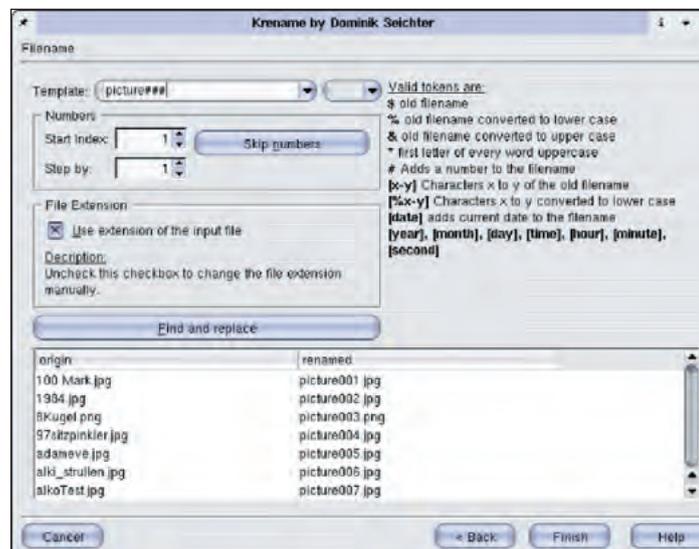
Desktop/Krename

Renaming individual files is easy from either the console or a window manager. Renaming a number of files from the console is sometimes fairly easy, depending on the changes you need. Renaming a whole load of files in various subdirectories, say to change filename extensions, the case of the names or to move them elsewhere, is generally far from simple. Unless, that is, you use *Krename*. This program offers a wide range of options, from simply renaming files from upper case to lower case, through changing filename extensions to numbering files, adding the current date to the name or changing file dates, ownerships or permissions.

Krename can also rename MP3 or Ogg Vorbis files based on the contents of the ID tags. there is a plugin API so that special functions for other types of files can be added. It is supplied as RPMs for the major distributions using them, as well as source, so it should be easy to get running on almost any system.

Desktop/Stellarium

Stellarium is a desktop planetarium. It



Make easy work of renaming batches files and directories with *Krename*.

renders 3D photo-realistic skies in real time with OpenGL. In addition to displaying stars, constellations, planets, nebulas and so on, it is also able to render landscapes, atmosphere and fog, to generate realistic looking night time land/skyscapes. *Stellarium* also contains information on 120,000 of the largest objects, taken from the

Hipparcos Catalogue, so you can click on the stars or planets to find out more about them.

Installation uses the standard `./configure && make && make install` method, but if you want to try it out first, skip the "make install" part and run *Stellarium* with `./src/stellarium`



The sky at night, even during the day. *Stellarium* brings the night sky to the warmth and comfort of your home.



Turn out the lights, and wish upon a star...

This keeps everything in the directory created when you unpacked the archive, which can be deleted if you decide *Stellarium* is not for you. Once you've run **make install**, the program can be run from anywhere.

Graphics/ImagePress

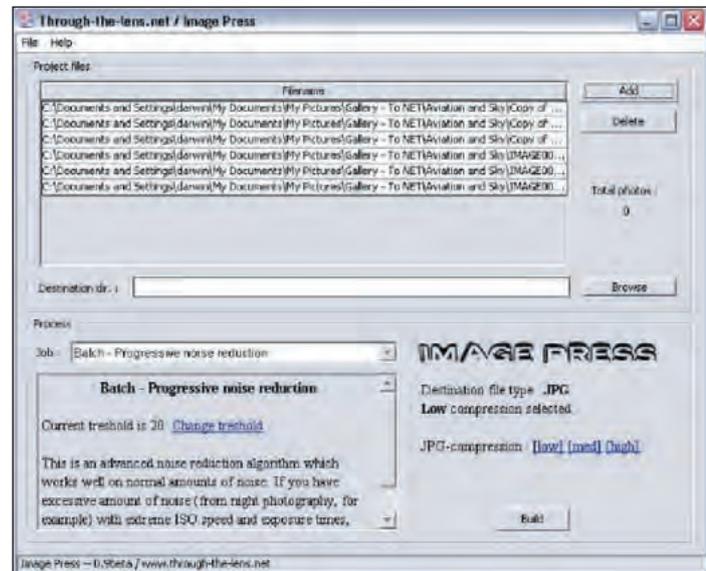
There are many programs for cataloguing the huge collection of images most digital camera owners amass, some of them have appeared on previous *Linux Format* coverdiscs. This is not one of them. *ImagePress* is

a batch image processing program, specifically written for use on digital camera pictures. It is particularly useful if you have taken a number of photos under less than ideal conditions. You can use *ImagePress* to apply noise reduction filters or exposure corrections to all the photos at once.

This is not an alternative to the many cataloguing programs out there, but a useful companion.

Internet/ ACPMoDem MwaveDriver

I have an IBM Thinkpad laptop which, like many other laptops, has a built in modem. These modems are generally "winmodems" and poorly supported under Linux. Fortunately for us, IBM are Linux-friendly and have released drivers for the MWave DSP/modem chip in many of their laptops. Don't rush to delete your Windows partition yet though, Windows can still perform



Process digital camera images in batches with *ImagePress*.

a useful function. It makes it easier to read details of the IRQs and other resources used by the modem, information you need to configure the modem driver. It is possible to get

this information from within Linux, but viewing the properties in the windows hardware manager is probably the easiest if the laptop has Windows installed. [LXF](#)

»» CD CONTENTS AT A GLANCE

Disc C

Magazine

HotPicks	All the programs covered in this month's HotPicks section
Java	Files for the Java tutorial
PHP	PHP source code and example scripts
ProcessAccounting	The files used in this article
WhatOnEarth	Files covered in the What on Earth feature

Desktop

ASmon	Afterstep or Window Maker CPU/Load/Mem/etc Meter
BubblingLoadMonitorApplet	Displays system load as a bubbling liquid
CUPSWordfilter	A MS Word document filter for CUPS
e3	A very compact editor
eXchaNGeR	An XML browser and XML editor framework
Fax4CUPS	Shell scripts that act as a CUPS backend
jMoneyAtHome	Financial software to manage the household
Kalculate	A simple calculator for KDE
Krename	A batch renamer for KDE
KSimus	Simulation and visualization of technical processes
LifeLines	Genealogy/Family history research tool
Minkowsky	A calendar, address book, and task management application
MTX	Control SCSI media changers and tape drives
MyCDCatalog	A tool that dumps CDROM information to a database
MySQLControlCenter	A GUI client for MySQL databases
Nano	Pico editor clone with enhancements
Openbox	A fast, slim window manager for X11 written in C++
OSX2X	A MacOS X utility for controlling a remote X session
Otak	A text-based visual interface to programs
PAI	A little AI program
PHPSambaExplorer	A PHP-based Web interface to smbclient
ProMio	A flexible, console-based menu program
RFStool	ReiserFS for Windows
ROXTasklist	A task/window list for the ROX desktop environment
ScriptingPluginForScribus	A scripting plugin for Scribus

SIDiBerner	A CD burner tool
Stellarium	A 3D astronomical sky renderer
Tipograf	A graphical a2ps frontend
TkPhone	A Tcl/Tk phonebook and calling system
uCON64	Handheld/console/emulation tool with too many options
Xdx	A GTK+ DX-cluster client for amateur radio
XFileExplorer	A file manager for the X Window System
Xvkbd	A virtual keyboard for X
Yudit	A Unicode text editor for the X Window System

Development

BlueJ	Java Development Environment aimed at teaching
Documancer	A programmer's documentation reader for X
Glib	The GLib library of C routines
GroupEnableClusterCompiler	Caches and distributes C/C++ compilation jobs
KDbg	Graphical KDE front end to the GDB debugger
MemCheckDeluxe	A memory usage tracker and leak finder
NASM-TheNetwideAssembler	80x86 assembler designed for portability and modularity
Octopus	C++ libraries for developing financial trading systems
PyChart	Python library to generate PS/PDF/PNG scientific charts
Python	A high-level scripting language
Twisted	Persistent, event-based framework for Internet applications
XPlanner	An extreme planning tool

Sound

AudioTagTool	An MP3 and Vorbis tag manager
CDRDAO	Disk-At-Once Recording of Audio CD-Rs
DigitalDJ	An SQL-based MP3 player frontend
DigitalMusicCenter	Music player supporting tvout with X10 remote
FMio	A small FM radio card manipulation utility
Funkyou	A fully-featured DJ program
HotLead	A utility for downloading full albums from Emusic.com
MpInuxman	File management for the MPMan F-60 USB MP3 player
Nogger	A bloat-free Ogg Vorbis player using GTK 2

Please note that the contents of the Games, Graphics, Internet, Mobile, Office, Server, System and Essentials directories can be found listed on CD C.

Mandrake 9.0



Neil Bothwick is your guide through the wonders of this month's jam-packed *Linux Format* coverdiscs. Starting, of course, with the long-awaited Mandrake 9.0, which features on the CDs and the DVD.

On the CD



Wherever you see this logo it means there's related stuff on the CD

Essential info

On page 107 we have grouped together essential info on the different types of packages on your coverdiscs – along with instructions for installing source packages.

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* coverdiscs are 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 coverdisc 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.

READ ME FIRST

Have we got a treat for you this month? CD users get three CDs, two containing the best parts of Mandrake 9.0 and another with our usual selection of goodies. DVD users get the full Mandrake 9.0 download edition, all 2GB of it, enhanced with the latest updates from Mandrake. Following the

success of using *jigdo* on the Debian DVD, we have also given DVD users the tools needed to create three ISO images, so they can install Mandrake on computers without DVD drives. There is a Windows version of *jigdo* too, so you don't have to run Linux in order to be able to install it. Despite being a complete distro,

Mandrake only occupies half of the DVD, there's still another 2GB of apps to play with.

We will look at the various ways to install Mandrake from DVD or CD, then take a tour of some of its highlights. Finally, we will have the usual round up of the contents of the other CD and the rest of the DVD.

Installing Mdk 9.0

You have a choice of three main ways of installing Mandrake 9.0. The easiest is direct from the DVD, which is what we will cover here. The other two are a network install or to create CD ISO images in order to burn CDs and install from them. Getting started with these two options is covered in the boxouts, but the main install process is the same whichever you choose. CD users simply boot from the first disc.

On booting from the disc, you will see the initial splash screen, where you can press **Enter** to proceed or **F1** to set options. For example you can change the resolution of the graphical install from its default of 800x600 to 640x480 or 1024x768 by typing **vgalo** or **vgahi** respectively. This is useful if you have an LCD monitor, which don't cope well with resolutions other than their default. You can disable the graphical install altogether with **text**.

You have a choice of a clean install or of upgrading an existing Mandrake system. At the same time you need to choose between the "Recommended" or "Expert" install modes. Expert gives you more choice, while Recommended makes many of the decisions for you. If you are new to Linux and have already created some empty space on your



Press F1 at the initial splash screen for extra installation options.

hard disk, you may wish to take the Recommended option. Most users with some Linux experience will probably prefer Expert, even if they don't consider themselves experts. Most of the choices have the same defaults as you would get with a Recommended install, but you do get the opportunity to make changes if you wish. We will walk through the Expert install here.

After selecting mouse and keyboard, you need to choose the security level. It is probably easiest to leave it at Standard here, although this

isn't really high enough if you are connected to the Internet, and change it in later in the *Mandrake Control Centre*.

The next stage is to create the partitions on your hard disk. In Recommended mode, this is taken care of automatically. To do the same in Expert, click the "Auto allocate" button in the *DiskDrake* window and select the "simple" option. This will create suitably sized root, home and swap partitions using the empty space on your drive. If you are comfortable setting up your own partitions,

Installing over a network

Harnessing transfer protocols

Mandrake doesn't allow you to export CDROMs (or DVDs) via NFS, but it is still possible to install over a network, providing the DVD equipped computer is running a web or FTP server. We'll take a quick look at how you do this with *Apache*. You'll need to be root to do this. Open `/etc/httpd/conf/apache.conf` in your favourite editor and add this line

```
include conf/mdk90.inc
```

Then type these lines into a new file and save it as `/etc/httpd/conf/mdk90.inc`

```
Alias mdk90/ /mnt/cdrom/
<Directory /mnt/cdrom/Mandrake>
    AllowOverride All
    order allow,deny
    allow from all
</Directory>
```

and restart *Apache* with

```
apachectl restart
```

You could have added the lines directly to *Apache's* configuration file, but this way is cleaner and means you can easily remove it by commenting out one line.

Now you need to create a boot disk using either `network.img` or `pcmcia.img` from the images directory of the DVD. The former file also includes drivers for some PCMCIA network cards, so it's a matter of trying each to see which works with your setup (my NetGear FA411 card needs `pcmcia.img`). Create the disk in the usual way

```
dd if=/mnt/cdrom/images/network.img of=/dev/fd0
```

Now boot from the disk, making sure

the DVD is mounted on the server computer. After giving the network information (hostname, IP address, DNS and gateway addresses) for the target machine, you'll be asked for the server details. Give the machine's IP address and use `"mdk90/"` for the path. The install will now proceed just as if you had booted from the DVD. The only difference is that you have already given network information, so you won't need to set this up later.

DiskDrake allows you to do this too. Click on "Done" to save the partitions and proceed to formatting. If you have no empty space, *DiskDrake* is able to resize Windows FAT32 (but not NTFS) partitions. See the tutorial on the first disc for more information.

Selecting packages

Now it's time to select the packages you wish to install. Tick the groups of packages you need, there may be some that aren't obviously necessary. You should select Development, even if you are not a developer, so you have all the tools needed to compile packages from source. If you have a network, select the Network Server group if you want to be able to share files or printers with other computers. There's no need to select everything "just in case", it is easy to install anything you later need from the



Disk partitioning – Windows partitions can be resized to make room for Mandrake, but read the instructions in the tutorial directory first.

Software Manager. If you have the CD edition of the magazine, make sure you untick the option for the third CD before package selection, otherwise it may try to install packages you don't have.

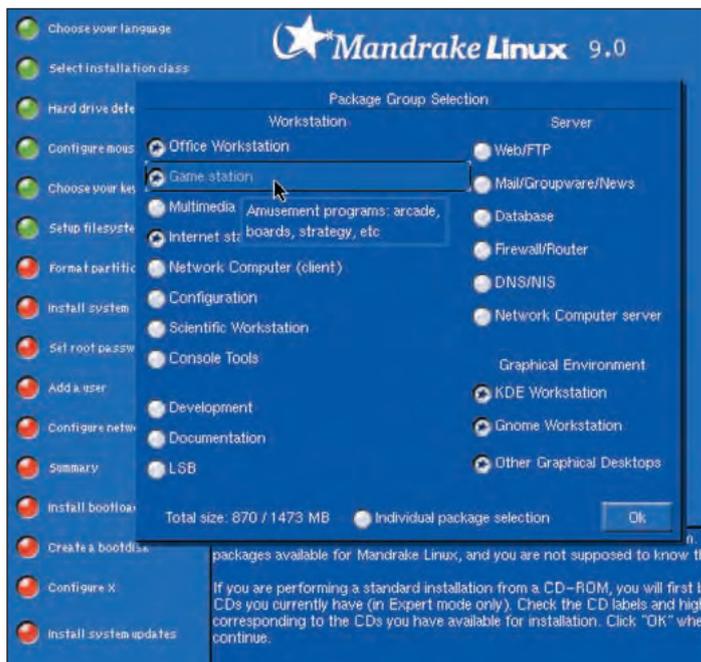
Now go and put your feet up while

the packages are installed, although CD users will have to pop back to change discs when asked. Package installation can take a while, then there is some basic config to do. The install creates an account for the root user and prompts

for a password, then you can create additional users. It is important to create at least one normal user, you shouldn't log in as root except when you need to.

You should normally accept the offer to set up a bootloader, this is what starts up Mandrake Linux when you boot. If you have Windows installed, the installer will create a menu to choose between it and Linux at boot time. If you decide to skip this step, you must create a boot disk at the next stage, or you won't be able to use your Linux setup. A boot disk is a good idea, even if you set up the boot loader. If you reinstall Windows, it will overwrite the bootloader and you'll need a way to get it back. A boot disk is one method, the other is to boot from the install CD/DVD, press **F1** and type **rescue** at the prompt.

If you set up networking or your Internet connection, you now have the opportunity to check for updates. If at



Choose packages in groups, based on your intended use of the computer...



... and you can fine tune that choice by adding or removing individual apps.

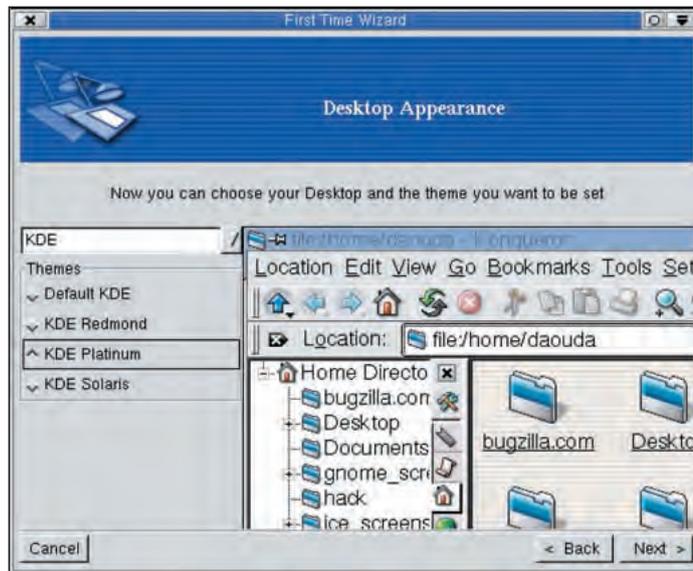
CoverdiscCD

« all possible, you should do this now. It not only makes sure your system is secure, it sets up the *Software Manager* for future updates, more on this later. The DVD version has already been updated with the latest versions available at the time, so there should be less to download for DVD users. This also applies to those who have created CDs from the DVD.

That's it, you're there. Remove the CD/DVD from the drive, insert your boot floppy if you didn't install a bootloader, and let the computer reboot. Over the next two pages, we'll take a look at some of the features of your new OS.

What've I installed?

Now that you have installed Mandrake 9.0, let's take a look around. The first time you start Mandrake, the *First Time Wizard* appears. This creates some basic settings and then collects info to register you with Mandrake's website. If you don't want it to send your personal information, or you have already done this on a previous install, you can press the cancel button at this point. Once the KDE desktop has loaded, click on the KDE menu button at the bottom



The *First Time Wizard* is run the first time you log into Mandrake (in case you hadn't worked that out). It sets up a basic default config for each user.

left of the screen. One glance should be enough to tell you that we won't be able to cover this in a page or two, so we will just pick out some highlights.

You can find further information in the tutorial directory of the DVD or first CD.

One of the first places you should

know about is the *KDE Control Centre*. It is probably not a good idea to go changing every setting you can find as soon as you've installed Mandrake, but it is useful to know that the options are here. LookNFeel has most of the options that relate to the appearance and

behaviour of the desktop. It is safe to experiment with the settings here, each panel has a Defaults button to go back to the original settings, and Reset to go back to the previously saved setting. It's not always necessary to start up the *KDE Control Centre* to make a change, many aspects of the desktop can be altered by right-clicking on them. Right click on the taskbar to change its settings, size or position. Do the same on the desktop to change its appearance, you can also drop picture files onto the desktop to instantly change the wallpaper.

For some reason, Mandrake have omitted the logout button from the panel in this release. While it's easy enough to logout from the KDE menu or by right-clicking the desktop, you can also put it back where it was in previous releases. Right-click on the panel and select Add->Applet->Lock/Logout applet from the popup menu. If you don't like where KDE has placed it, right-click on the bar to the left of the button and select move, slide it to where you want and left-click.

Tasks, not programs

The programs accessible from the KDE

D.I.Y. ISO images

jigdo for Mandrake

The subject of whether DVDs should contain bootable distro installers or ISO images has come up again on our web forums. A bootable DVD is clearly better if you want to install the distro to a DVD equipped computer, it saves the hassle and time of burning the ISO images to discs and means you can leave the installer to get on with installing all the packages you want from the distro without having to be there to change

discs. However, if you have more than one computer and one doesn't have a DVD drive, ISO images have an advantage, especially if that computer isn't networked to a DVD equipped machine.

We ran a survey on the website to see which you preferred, bootable DVDs or burn-it-yourself ISO images, and the split was almost exactly 50:50. Including both on the DVD is not an option,

because it would leave no room for anything else, so we have set this DVD up to be bootable, and added the option to create ISO images if you need them. Before you use this, consider the alternative of a network install, see the separate boxout on this. It really is a much easier option.

Mandrake provide their own means of building ISO images, but this requires that you be running Linux before you can use it. This is fine for those updating, but not for first time users. So we have used the *jigdo* system that was originally developed for use with Debian, and was used on our Debian cover DVD. There are Linux and Windows versions of *jigdo*, making it suitable for all users.

Creating the ISOs in Linux

We've added a script to make life easier this time. First you need to copy the *jigdo* directory from the DVD to a partition with enough space to store the ISO images, 2GB if you want to create all three at once. Drag it from the DVD using your graphical desktop or type

```
cp -a /mnt/cdrom/jigdo /path/to/dest/
```

Now change to that directory and run the *mkiso* script

```
cd /path/to/dest/jigdo
```

```
sh mkiso
```

This will create all three ISOs. You can create individual ones by giving the numbers as arguments to *mkiso*. To create the first two, for example, type

```
sh mkiso 1 2
```

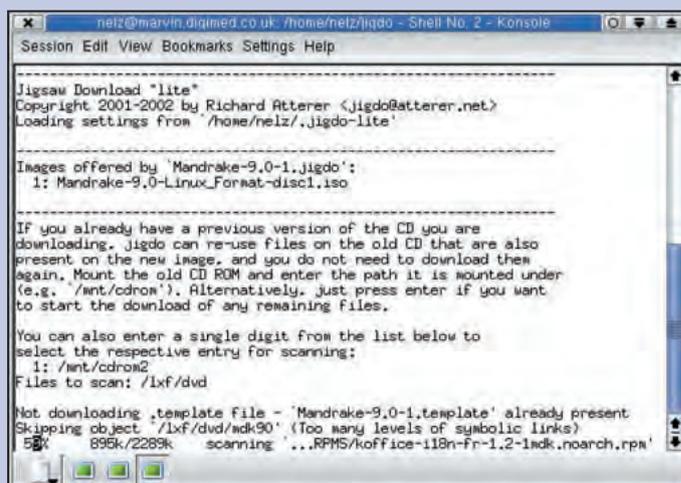
Scanning the DVD for the first pass can take quite a long time, because the whole DVD has to be scanned. The second and third discs are built much faster. If you do want to use Mandrake's *MakeCD* script instead of *jigdo*, simply change to the directory in which you want to create the ISO images and type

```
/mnt/cdrom/misc/MakeCD  
-t . -a /mnt/cdrom/Mandrake
```

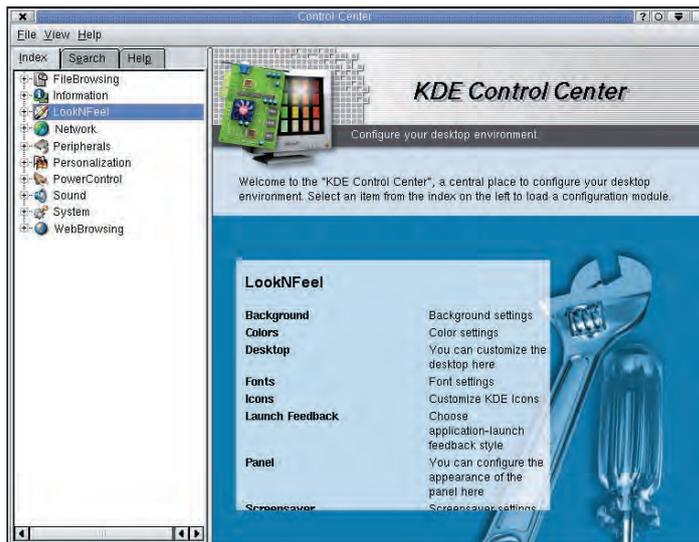
Creating ISOs in Windows

Copy the *jigdo* directory to your hard drive, making there is enough space for the ISO images, and double-click the *jigdo-lite.bat* icon. Then follow the prompts. There are three images to create, so you'll have to run this for each of the three *jigdo* files. There are reports of *jigdo* not working on Windows ME, although we have not been able to test this.

You now have three ISO images that you can burn to CD in the usual way.



Creating CD ISO images from the DVD is easy with *jigdo*.



KDE Control Centre lets you change just about any aspect of KDE's operation.

menu are grouped according to function, such as Multimedia, Office, Configuration and so forth. Under these grouping you will find the programs listed by name, which is fine if you know the name of the program that does the job you want done. The "What to do?" menu item contains entries for common tasks. For example, to play MP3s select "Enjoy music & video->Listen to music files". This starts the *XMMS* music player, just as if you had run it from the Multimedia->Sound menu, but it didn't require you to know that the MP3 player is called *XMMS* (would you have guessed that was the name if you didn't know?). The *What to do?* menu should keep you occupied for some time, exploring the various options and programs available.

One of the KDE Desktop's most useful features is hidden away in the docs. Press **Alt-F2** and you'll get a small window where you can type a command, file, path or URL. *Konqueror* will open a window displaying this, whether it is a webpage, FTP site or local directory. It maintains a history of locations typed here, so you only need to type the first few characters the next time. If you type **#** followed by the name of a command, such as **#rpm**, you will be shown the man page for that command, formatted as a webpage.

Konqueror is able to handle several other types of "file". If you don't have a column of icons down the left hand side of the *Konqueror* window, press **F9** to add it. Then click on the Services icon, second from the bottom. Put an audio CD in the drive and select Audio CD Browser. The tracks on the CD are listed in the main window as though they were audio files. Open the Ogg

Vorbis folder and you'll see the tracks listed as if they were Ogg Vorbis files. Copy one elsewhere and it will be encoded as it's copied.

Neighbours

A question often asked by people moving from Windows to Linux is "where is the *Network Neighbourhood*"? The answer is that it is just below the Audio CD Browser, called LAN Browser. You may need to do some config before you can use it. Firstly, check that you have a package called *lisa* installed. Either use the Software Manager or type **urpmi lisa** in a console. Then start up the *KDE Control Centre*, go to Network->LAN Browsing and click the "Guided LISA Setup" button. After you have gone through the settings, click Apply. Now you can use the LAN browser to view

Samba shares and FTP servers on your network.

During installation, you were given the option to have one user logged in automatically on startup. This is a convenient feature, although it is less secure if others have physical access to your computer, as they can switch it on and be logged in as you. This, and many other items, can be controlled from the *Mandrake Control Centre*. This is not to be confused with the *KDE Control Centre*, it is a completely different program, launched from the KDE menu or an icon in the taskbar. The *Mandrake Control Centre* has to be run as root, so you'll be asked for the root password. To change the automatic login, click on the System icon and then Users. Here you can make various changes, including enabling and disabling automatic logins. The *Mandrake Control Centre* is also the place to configure networking, hardware or the display resolution.

Software manager

Towards the end of the installation process, you were asked to select a server from which to download updates. In addition to being used at that time, the server was also added to the list of sources used by Mandrake's *Software Manager*, *rpm-drake*. The other sources in there are the installation media. This makes installation of further software or dependencies from Mandrake's sources much easier. For example, let's say you didn't select *MySQL* when installing Mandrake but now need it. There are several dependent packages involved, which would each need to be

sourced an installed using the standard **rpm** command. With Mandrake, you can do it in one step, either from a GUI or the command line. The Software Manager GUI is started from the KDE menu at Configuration->Packaging->Install Software. Type **mysql** in the search gadget to get a list of matching packages. You only need to select the main *MySQL* package, *rpm-drake* will sort out the dependencies. To install from the command line, type

```
urpmi mysql
```

It gets better, because you can add extra sources to the list. This is done from K->Configuration->Packaging->Software Sources Manager or by using **urpmi.addmedia** from the command line. If you are a member of Mandrake Club you will have access to newly created RPMs, before they go in to the public contrib area, the Mandrake Club website tells you how to add this source. The contrib area contains packages that are useful, but not part of the main distribution. You can add this by typing

```
urpmi.addmedia -h Contrib ftp://
```

```
ftp.mirror.ac.uk/sites/sunsite.uio.no/
```

```
pub/unix/Linux/Mandrake/Mandrake/
```

```
9.0/contrib/RPMS
```

This is for the main UK mirror, you can find your local mirror from the Mandrake download page at www.linux-mandrake.com/en/ftp.php3. You will usually need to go up a directory from the link here to find the contrib directory. Now let's add another:

```
urpmi.addmedia plf http://
```

```
www.zoreil.com/mirrors/www.plf.org/
```

```
9.0/i586 with ../hdlist.cz
```

Once again, you'll need to visit the site to find the best mirror to use, and it gives the **urpmi.addmedia** command for each mirror so you can paste it instead of typing. The Penguin Liberation Front (PLF) contains packages that could not be included in the Mandrake distribution for legal reasons, often relating to US law. This includes packages like *libdvdcss*, needed to read CSS encrypted DVDs.

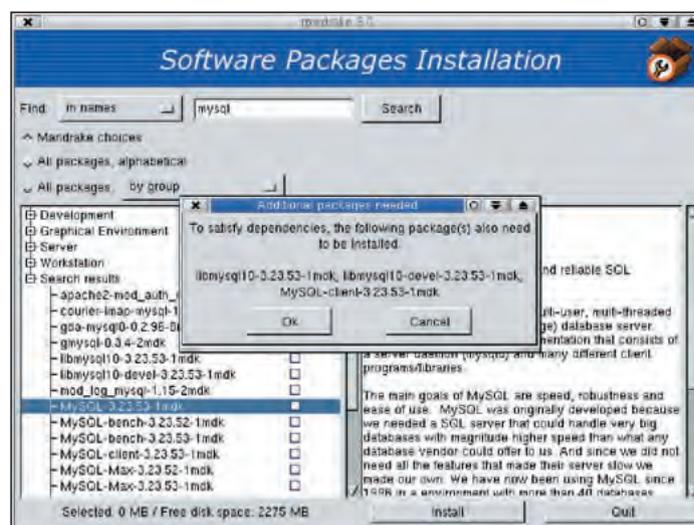
It doesn't matter which site has a package, **urpmi/rpm-drake** takes care of that, so typing

```
urpmi dvdcss
```

will locate and install the packages needed to read CSS encrypted DVDs. Lists of the available packages are stored locally, so you will need to keep them up to date with

```
urpmi.update -a
```

either from the command line or run from *cron*. **LXF**



The Software Manager (*rpm-drake*) takes care of installing all package dependencies, even if they are located on a different server.

User Groups

LUGs worldwide are full of members keen to help with your problems, discuss ideas, and generally natter about all things Linux. You can find lots more information online at: www.lug.org.uk

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

4 Oxford

URL www.oxford.lug.org.uk
Contact Alasdair G Kergon

5 Kent

URL www.kent.lug.org.uk
Contact John Mills

6 Brighton

URL www.brighton.lug.org.uk
Contact Johnathan Swan

7 Worcestershire

URL www.worcs.lug.org.uk
Email info@thirdeyedevlopment.com

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.wolveslug.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/>
Contact John Southern

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.wy.lug.org.uk
Contact Jim Jackson

27 Sheffield

URL www.sheflug.co.uk
Contact Richard Ibbotson

28 Staffordshire

URL www.staffslug.org.uk

29 North East

URL www.shofar.uklinux.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.sderbylug.org.uk
Contact Dominic Knight

36 Belfast (BLUG)

URL www.belfastlinux.cx
Email russell@belfastlinux.org

37 Wiltshire

URL www.wiltshire.lug.org.uk
Contact Jason Rudgard

38 South London

URL www.sl.lug.org.uk
Email ben@ilovephilosophy.com

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 <http://midlandslug.port5.com/>
Contact Pete Thompson

42 Cumbria

URL www.cumbria.lug.org.uk
Contact Jamie Dainton

43 Dorset

URL www.dorset.lug.org.uk
Contact John and Mat

44 Shropshire

URL www.shropshire.lug.org.uk
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 – see 87

URL www.kemputing.net/lug/anlug-aims.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

51 St Albans & Luton

URL 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

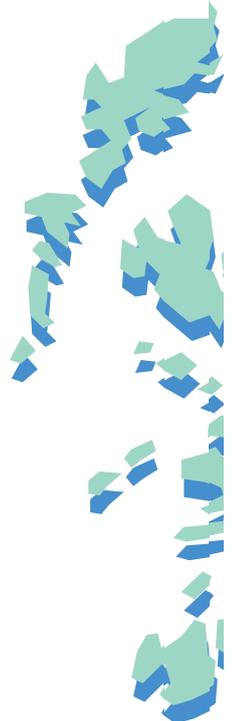
URL www.preston.lug.org.uk
Contact Phil Robinson

54 Derry

URL www.derry.lug.org.uk

55 Isle of Wight

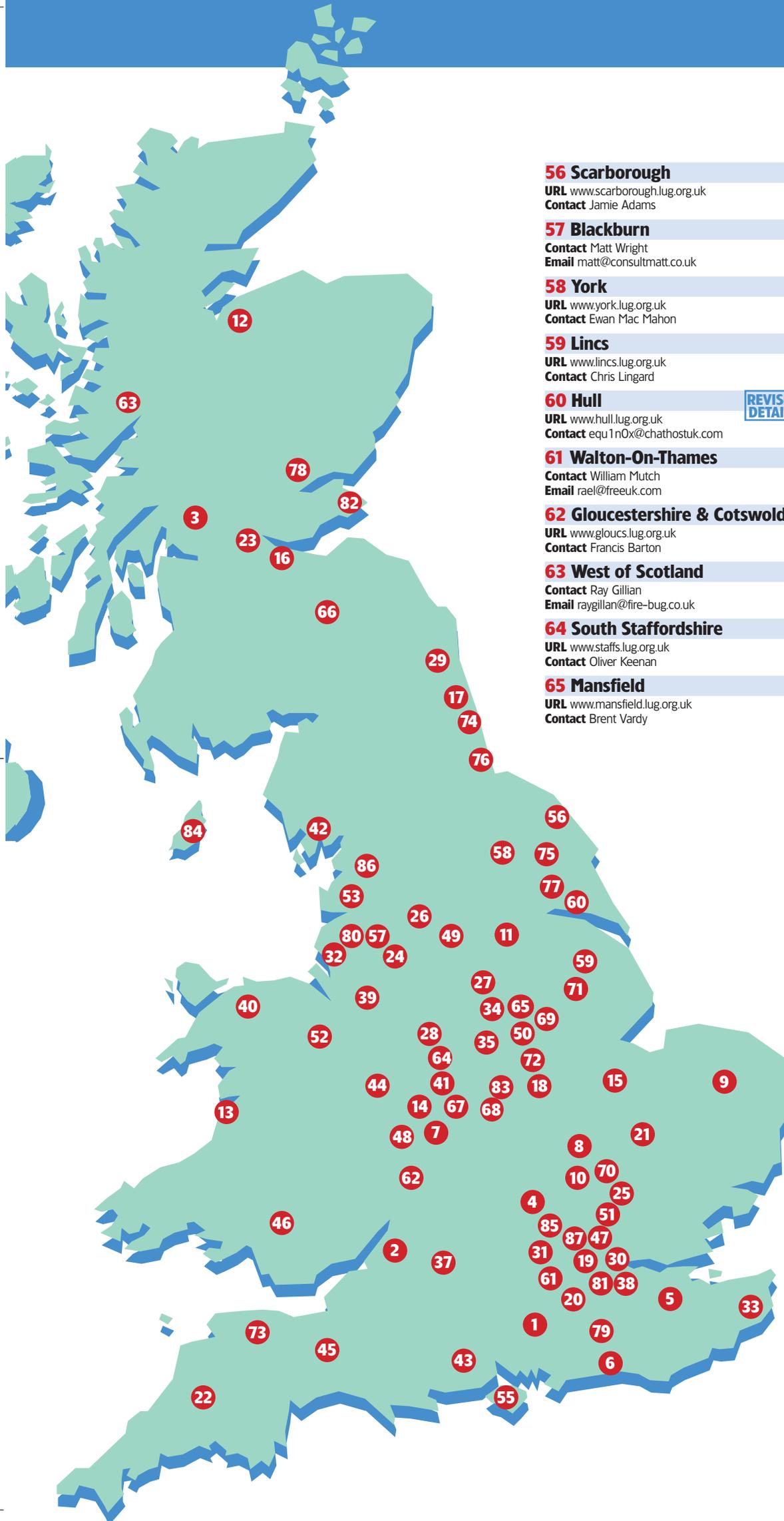
URL www.iow.lug.org.uk
Contact David Groom – info@iow.lug.org.uk



54

36

REVISED
DETAILS

**56 Scarborough**

URL www.scarborough.lug.org.uk
 Contact Jamie Adams

57 Blackburn

Contact Matt Wright
 Email matt@consultmatt.co.uk

58 York

URL www.york.lug.org.uk
 Contact Ewan Mac Mahon

59 Lincs

URL www.lincs.lug.org.uk
 Contact Chris Lingard

60 Hull

URL www.hull.lug.org.uk
 Contact equ1n0x@chathostuk.com

REVISED
DETAILS**61 Walton-On-Thames**

Contact William Mutch
 Email rael@freeuk.com

62 Gloucestershire & Cotswolds

URL www.gloucs.lug.org.uk
 Contact Francis Barton

63 West of Scotland

Contact Ray Gillian
 Email raygillian@fire-bug.co.uk

64 South Staffordshire

URL www.staffs.lug.org.uk
 Contact Oliver Keenan

65 Mansfield

URL www.mansfield.lug.org.uk
 Contact Brent Vardy

66 Borders

URL www.linux.bordernet.co.uk
 Contact Welby McRoberts

67 South Birmingham

URL www.sb.lug.org.uk
 Contact Tim Williams

68 Coventry

Contact Darren Austin
 Email info@coventry.lug.org.uk

69 Newark & Lincoln

URL www.newlinc.lug.org.uk

70 Bedfordshire

URL www.beds.lug.org.uk
 Contact Neil Darlow

71 Lincoln

URL www.lincoln.lug.org.uk
 Contact Jon Shamash

72 Loughborough

URL www.loughborough.lug.org.uk
 Contact Martin Hamilton

73 Exeter University

Contact Nicholas Murison
 Email N.J.Murison@exeter.ac.uk

74 Sunderland

Contact Thomas Croucher
 Email thomas.croucher@sunderland.ac.uk

75 East Yorkshire

Contact Daniel Gallacher
 Email sharkonline@whatemail.com

76 Cleveland Open Source Group

Contact Haniff Din
 Email openlug@digitalmedia.co.uk

77 Beverley

Contact Vladimir Lukyanov
 Email vladimir_lukyanov@hotmail.com

78 Dundee & Tayside

URL www.dundee.lug.org.uk
 Contact Duncan Gauld

79 Sussex

URL www.phpworld.co.uk/~swlug
 Contact Gareth Ablett

80 Wigan & St Helens

Contact Paul F Johnson
 Email paulf.johnson@ukonline.co.uk

81 Brixton

URL www.communitytechnology.org.uk/~linuxhome
 Contact R M Sanchez

82 St.Andrews, Fife

URL www.standrews.lug.org.uk
 Email stuart@nx14.com

REVISED
DETAILS**83 Nuneaton**

URL www.nuneaton.lug.org.uk
 Contact S Prosser

84 Isle of Man

URL www.iom.lug.org.uk
 Email helix@manx.net

85 Aylesbury

URL www.aylesbury.lug.org.uk
 Email drbond@educational-computing.co.uk

86 Lancashire

URL www.lancasterlug.org.uk
 Contact Jonathan Hill

87 North London – see 47

Contact Jason Hodson
 Email jason@voyagercomputers.co.uk

LinuxUserGroups

LUG OF THE MONTH!

Kingston upon Hull

Hull LUG started towards the end of 2001, it was then purely a web-based LUG, using a news and forum system hosted by a new local web hosting company. (Matrixhost.co.uk). This was met with enthusiasm by the community and soon we had in excess of 20 members on the site. It was then that we held our first meeting, which was incidentally an online IRC meeting. This was not so well supported as the members on the

site, but nevertheless had around 10 members talking Linux.

The first real meeting was mid-2002, in the local pub. This was well attended, with at least a dozen members. This is now a monthly event: plans for a wireless networking project and various Linux World Domination techniques are discussed. Members bring their laptops and distros for swapping, and each other's problems are discussed and

solved as well as the usual Linux banter and fun.

If you would like to join us for a meeting the place is the *Old Grey Mare* in Hull, 8pm, first Tuesday in the month (Look for the lot with laptops :). If you don't know where that is, there is the mailing list to ask on, or simply join us in IRC and ask one of us on [#hull](http://irc.chathostuk.com).
www.hull.lug.org.uk



Worldwide Linux User Groups

Free Software users across the globe

Africa

EGYPT

URL www.linux-egypt.org
Contact Hesham Bahram

GAUTENG, SOUTH AFRICA

URL www.glug.org.za
Email glugmin@revolution.org.za

Australia

ADELAIDE

URL www.linuxsa.org.au
Email mtippet@anu.edu.au

ALICE SPRINGS

URL www.aslug.org.au

MELBOURNE, VICTORIA

URL www.luv.asn.au
Contact luv-committee@luv.asn.au

PERTH

URL <http://plug.linux.org.au/>

SYDNEY

URL www.slug.org.au

Europe

AUVERGNE

URL www.linux-arverne.org
Email Cyril.Hansen@wanadoo.fr

COSTA DEL SOL (English speaking)

URL www.fuengirola.lug.org.uk

DENMARK

AIssund www.alslug.dk
Esbjerg www.eslug.dk
Fyns www.flug.dk

Midt-og Vestjylland www.mvjlug.dk

Nordjylland www.njlug.dk

Skåne Sjælland www.sslug.dk

Trekantsområdet www.tlug.dk

Vest-fyn www.haarby-net.dk/vflug

Århus www.aalug.dk

EIRE

URL www.linux.ie
Email root@linux.ie
URL www.dilu.org
Email glossary@dilu.org

GOTHENBURG

<http://nain.oso.chalmers.se/LUGG/>

India

URL www.linux-india.org
Email newsmaster@linux-india.org

TRIVANDRUM

URL www.river-valley.com/tux
Email anil@river-valley.com

North America

ALASKA

URL www.aklug.org
Email deem@wdm.com

ALBERTA

URL <http://calgary.linux.ca/>

BATON ROUGE

URL www.br lug.net
Email dpuryear@usa.net

BAY AREA

URL www.balug.org
Email aftyde@balug.org

CLARKSVILLE, TN

URL www.cllug.org
Email tux@cllug.org

DENVER

URL <http://clue.denver.co.us/>

FLORIDA

URL www.flux.org

LOS ANGELES

URL www.lalugs.org
Email dank@alumni.caltech.edu

NORTH COLORADO

Email nclug@nclug.org

OTTAWA CANADA

URL www.oclug.on.ca

TAMPA

URL www.suncoastlug.org
Email president@suncoastlug.org

UHACC Normal, IL

URL www.uhacc.org
Email lug@uhacc.org

VIRGINIA TECH

URL www.vtlug.org
Email nega@vt.edu

South America

BUENOS AIRES

Email dcoletti@impost.com.ar

CHILE

URL www.linux-chile.org

MONTEVIDEO

URL www.linux.org.uy

PARAGUY/ ASUNCION

Email rolgiati@conexion.com.py

SAO PAULO

URL <http://gul.ime.usp.br/>
Email gul@ime.usp.br



In advocating the use of GNU/Linux, says **Jono Bacon**, presentation is the key.

One of the most important factors in the transmission of info from one head to another is presentation. The presentation of even the dullest of content can make a huge difference. This is no different in the Linux advocacy world, and if you don't present your ideas clearly, you are not going to win many people over.

One of the key problems with many advocacy documents and resources is that they are written by techies. While this is fine to an extent, many of the advocacy articles and documents assume prior knowledge; and that assumption can be faulted. It is always important to try and visualise the context that the punter is in, and to weave together a solution that is both pragmatic and flexible. In many cases you need to be blunt about how this benefits the punter, and that the migration and learning curve of Linux are well worth it.

Another essential ingredient to good presentation is to be clear and focussed and not to stutter or slur on your ideas.

The punter may try to play with you to test your argument, but if you are aware of what they need, what Linux has to offer and how you can help them get it, you will have few problems. Remember that people are cynics – they may ask what the catch is in an OS that does everything they want and more at no cost. Also remember to always be honest – if you lie about Linux, you do more of an injustice than a justice.

The core aspect of creating a well presented view of your ideas is to make your solution viable for the punter. A small business for example would not be able to install a Beowulf cluster, but a big business may. Identify your users' needs and produce a plan that fits into their way of doing things. If you develop your ideas carefully and present a structured knowledgeable argument, you will attract interest from many people.

Next month we will begin taking a look at how we can best write and formulate our ideas in document form, and look at the issues that are involved in writing a good article or paper.

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**Production Editor** Richard Smedley richard.smedley@futurenet.co.uk**Editorial Contributors**

Jono Bacon, Neil Bothwick, Chris Brown, David Cartwright, Andy Channele, David Coulson, Richard Drummond, Hoyt Duff, Keith Gilbertson, Paul Hudson, Maurice Kelly, Neil Lucock, Charlie Stross, Paul Tansom, Adam Vaughan

COVER CD PRODUCTION**CD Editor** Neil Bothwick**Senior CD Editor** Kate Hadley**ART CONTRIBUTORS****Photography** James Wilson, Louise Parker, Photodisc**Illustration** Paul Bateman, Chris Winn, Shane Collinge**ADVERTISING SALES****Portfolio advertising manager** Matt Stanley 01225 442244matt.stanley@futurenet.co.uk**Portfolio deputy ad manager** Diane Clydesdale**Business development manager** Damian Hughes**Account managers** Zoe Drew, Giles Crosthwaite Scott,**Ad sales executives** Jo Doyle, Stephen Hall**MARKETING AND PROMOTIONS****Marketing Manager** Fiona Tully**Subscriptions Executive** Marie Spicer**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 LicencesSimon 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 coverdisc demos and reader submissions are supplied to us on the assumption they can be incorporated into a future covermounted disc, 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).

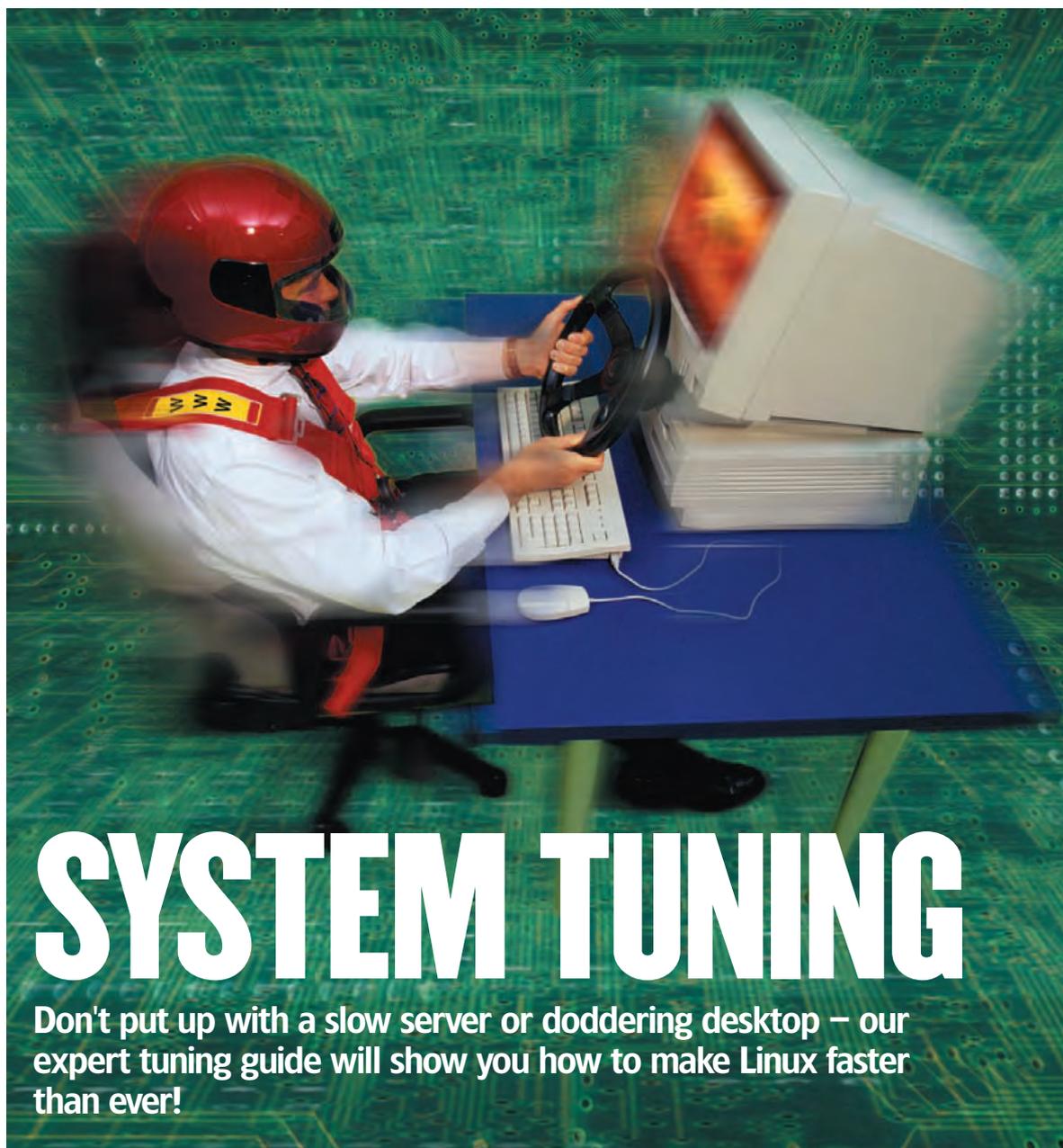
Non-executive Chairman: Roger Parry**Chief Executive:** Greg Ingham**Chief Operating Officer & Managing Director, UK:** Colin Morrison**Group Finance Director:** John Bowman**Tel** +44 (0)1225 442244

Media with passion

www.thefuturenetwork.plc.ukBATH • LONDON • MILAN •
NEW YORK • PARIS •
SAN FRANCISCO • WROCLAW

NEXT MONTH

Issue 36 on sale Monday 30 December



SYSTEM TUNING

Don't put up with a slow server or doddering desktop – our expert tuning guide will show you how to make Linux faster than ever!

INSTANT MESSAGING

You may be hooked on IM, but which is the best client? Find out with our extensive roundup

PLUS: Reviews of the Jool Kwartz server, Xandros, RealSoft 3D, Webmin, Arkeia 5 backup and more!

AND: Don't forget we'll have two packed CDs or a DVD, and another unmissable Linux Pro mini-mag!



The exact contents of future issues are subject to change

DON'T MISS YOUR COPY OF LINUX FORMAT!

Get it delivered to your door every month – subscribe on page 96

LINUXPRO

From the makers of LINUX Format

Christmas 2002

ACCESS RIGHTS

What your business
needs to know about
accessibility requirements

PLUS

The Blender Story

How Open Source saved 3D!

Multiprocessing

Parallel programming –
when and how to use it

Contacts

EDITORIAL

Editor

Nick Veitch
nick.veitch@futurenet.co.uk

Art editor

Julian Jefferson
julian.jefferson@futurenet.co.uk

Production editor

Richard Smedley
richard.smedley@futurenet.co.uk

Contributors

Biagio Lucini, Richard Smedley,
Nick Veitch

Photography

Corbis, Powerstock

Publisher

Dave Taylor

ADVERTISING

Portfolio ad manager

Matt Stanley
matt.stanley@futurenet.co.uk

Portfolio deputy ad manager

Diane Clydesdale
diane.clydesdale@futurenet.co.uk

Business development manager

Damian Hughes
damian.hughes@futurenet.co.uk

Account managers

Zoë Drew,
Giles Crosthwaite Scott

Ad sales executives

Jo Doyle, Stephen Hall

LinuxPro is a part of



magazine

Published by



Media with passion

30 Monmouth Street,
Bath BA1 2BW
Tel +44(0) 1225 442244

Welcome

Twenty-four pages of real-world Linux for IT professionals

It may be the yuletide season

but we've still been hard at work bringing you another issue of *Linux Pro*. One of the topics we are covering this issue has been planned for quite a while now, though it may be something you and your IT department have given little thought to – accessibility.

With legislation already in force in some countries and being actively considered in others, it should be something you are taking seriously. Besides, providing access to your IT services to, for example, visually impaired people, may give your company the chance to employ some gifted employees. In the first part of this feature we look at some of the problems and the solutions which already exist, plus the likely extent of your obligations.

If you're more concerned with getting more from your hardware than your employees you should take a look at our High Performance Computing feature,



which starts off looking at why you don't need expensive hardware to indulge in parallel programming.

To round off this issue, find out how some great 3D software was literally saved from oblivion by the Linux community.

Nick Veitch Editor
nick.veitch@futurenet.co.uk

CONTENTS

High performance computing:

Two processors is a good start **p4**



Access rights: your obligations on accessibility **p10**

Blender3D Saved by enthusiasts! **p18**



FEATURE | HIGH PERFORMANCE COMPUTING

HIGH PERFORMANCE COMPUTING

In the first of a short series, **Biagio Lucini** explains parallel programming, and when it is appropriate to resort to it.

Parallel programming is often seen as a universal panacea. While this is the chances are that when executed in parallel on more than one CPU (if common application will run faster. In this article we discuss when it is parallel programming, together with a few issues related to it.

Supercomputers are one of the most evident examples of how sometimes reality looks more incredible than pure imagination. As a matter of fact, when in the late forties the first computers were built, nobody could imagine that they would have become so important a part of our life. This is mainly due to the ability of a modern microprocessor to perform an incredible number of operations in an incredibly short time (e.g., for a 1.5GHz Pentium IV we are roughly talking about one billion operations per second), together with affordability. Supercomputers are a quest for performance pushed to its limits and possibly beyond: the most modern supercomputer, hosted at the Earth Simulation Centre at Kanazawa, Japan, has a quoted performance of around 38 trillion operations per second.

But supercomputers are not just fashionable because they can stimulate our imagination: while not as cheap as standard desktop machines, they are not anymore out of reach of medium-sized companies. Probably one of the key factors that have made this possible is the Beowulf project. Started as an investigation aimed at exploring alternatives to the very expensive traditional supercomputers, the Beowulf project has now evolved so much that the fastest Beowulf cluster at the time of writing ranks fifth in the classification of the most powerful computers in the world (www.top500.org). What has made the Beowulf project possible, apart from the increasing availability of inexpensive but efficient hardware, is Linux, an operating system that offers stability and reliability at a very low, if not null, cost.

In the most simple approach, a Linux-driven supercomputer can be seen as a bunch of

computers able to communicate and co-operate. To this category belong not just Beowulf clusters, but also the often less expensive SMP (Symmetric Multi Processor) machines. In this short series of articles we are going to investigate how to get very high performance from machines with more than one CPU (as opposed to single CPU systems, in jargon referred to as scalar machines), with emphasis on SMP machines. In particular, this first piece will give an overview of general problems and aspects related to cooperative

“Supercomputers are a quest for performance pushed to its limits, and possibly beyond them”

multi processor machines. It will serve both as an introduction to parallel programming (the main subject of this series) and as an exposition of the possible benefits (weighted against the disadvantages) associated with it. We won't assume any prior knowledge of the subject.

As far as hardware issues are concerned, we won't go into much details: we will just mention the few most common examples, with the aim of helping people to choose the best system for their needs. Knowledge of a few coding paradigms will be a desirable prerequisite, although, especially in this first part, we are going to use simple and intuitive examples. The programming language of choice will be C, but we will also translate some of the examples to Fortran, as an explicit acknowledgement of the importance of this language in the high performance world.

Myths and facts

Parallel programming is the art of taking advantage in the most efficient way of the presence of different processors. It requires that the developer makes the code aware of the topology of the system. Usually libraries



**is not always the case,
CPU (if possible) a
it is beneficial to resort to**

FEATURE | HIGH PERFORMANCE COMPUTING



are used for this purpose. Generally speaking, parallel programming is not easy and will probably be a big investment, for novices, in terms of time. In addition, it does not always give the wanted results in terms of performance. For this reason, before discussing parallel programming it is worthwhile looking at alternatives.

An often overlooked feature of most modern processors is that they can execute more than one instruction per clock cycle. This can be exploited by coding (part of) the program using some special set of instructions. The details vary from processor to processor, but the idea is the following. Suppose you have the following code:

```
for (i=0;i<100;i++){
  a[i] = b[i] + c[i];
}
```

What happens when **i=1** is independent from what happens for another value of **i**. This means that the sums can be executed at the same time. A good compiler is supposed to take care of generating efficient code for a given architecture, and this should include special instructions for concurrent execution of instructions with multiple data, but experience shows that this is not often the case, so human intervention is practically always needed. The speed improvement that can be

gained in this way depends on the processor type. For Pentium IV processors an improvement of a factor of five can be obtained. A Playstation 2 (with the Linux expansion kit) in turn should be able to perform five times faster than a Pentium IV when the application is explicitly coded with specific optimisations. In the latter case, we are already talking about 5.5 billion operations per second. If this is good enough for you, there is no need for parallel programming. However, it must be considered that this approach suffers from a few limitations: the program, or at least its numerically most intensive part, must be coded in a low level language; portability of the source code is lost, since the special instruction set depends on the architecture; memory access can be a severe bottleneck.

Performance gains

One of the most common beliefs is that in a multi processor system (MPS) the total power is just the sum of the powers of the single components. This, for instance, would mean that if, e.g., *OpenOffice.org Writer* takes ten seconds to load on my PIV 1.7GHz system, it will take one second on a multiprocessor system built out of ten systems like mine. Many people will be disappointed when after buying an expensive MPS, they see their favourite app not running any faster than on the single processor! This is because, surprising as it might seem to many, the application is still running on a single processor – because the application is not aware of the fact that there are many processors, and there is no simple way of spreading the load over several processors with external commands. All the kernel can do, e.g. in an SMP system, is to address different processes to different processors. In this way, what is achieved is load-balance, not high performance. Those who have decided that this will solve their problems can stop reading here if they wish. Instead those who really need more performance for a single application have to get their hands dirty in the source code, and in the worst cases even redesign the application. Even so, the chances are that the execution time does not scale with



the number of processors: apart from rare cases, different parts of the code assigned to different processors will need to communicate, and this will be reflected in a loss of performance. A good parameter of choice is to contrast the time needed to recode the application and the performance loss, with respect to the ideal case, with possible benefits.

Let us clarify the above issue with a simple example, which is very common in scientific computations (incidentally, in this topology or similar ones fall applications like meteorology, and in general all types of problems that require numerical solution of differential equations). Suppose that our program implements a bi-dimensional grid with $n \times n$ points, where at each point is associated a set of variables, and the calculation at a given point is influenced by the value of the variables in the nearest neighbour points, according to some formula. If we have a system of $m \times m$ processors, the obvious thing to do is to allocate a subgrid $(n/m) \times (n/m)$ on each processor. The number of communications a single processor has to perform involves only the nodes on the boundary of a subgrid, which are $4 \times (n/m)$. If n/m is a large number, quite likely the time spent in communication is negligible with respect to the time spent in computations within a given subgrid. If this is not the case, the computation will be dominated by communication time, and we could have been better off by allocating a fraction of the total number of processors to this particular problem and assigning the remaining processors to some other problem. For a small enough grid it is also possible we could have dealt with our system on a single processor, if nothing else (like, e.g., insufficient memory) would have prevented us from doing so. As a rule of thumb, usually one defines the efficiency on a system with p processors as $e = t(p)/(p \cdot t(1))$ where $t(p)$ is the execution time on the p processor system and $t(1)$ is the execution time on a single processor. Whenever the efficiency is close to 1 it is worthwhile to run the program on p processors; as soon as the efficiency drops below 0.5 one should

consider reducing the number of processors dedicated to the program.

Who needs parallel programming?

If you are still reading, then there is a chance that you can seriously benefit from parallel programming. To clear any possible doubt, here is a list of the most common cases for which parallel programming is likely to be appropriate: a) whenever the efficiency “ e ” is close to or equal to 1, the more the processors are involved the better;

“Many will be disappointed to see their favourite app not running faster than on a single processor”

b) whenever a program is trivially parallelisable (i.e. does not require communication at all, and in this case the efficiency is exactly 1 – although this is an ideal case);
c) whenever the execution time of the serial program is more than the average up time on a typical serial machine (since every machine needs some maintenance, we should take into account that at some point the machine could be switched off);
d) whenever the execution time on a scalar machine is bigger than the time by which the answer is needed (think for instance of weather forecast: a prediction for tomorrow will be useless if it arrives the day after tomorrow);
e) whenever the memory requirements of the application do not fit the size of the memory available to a single processor;
f) whenever the speed at which data are fed to the processor is a severe bottleneck.

A common prerequisite is that the code is such that there are adequately large mutually independent portions (or the code can be rewritten in that way). Those people with this prerequisite who recognise themselves in one of the above cases should seriously consider parallel programming.





There is another category of people that may want to consider parallel programming: those who in principle do not need it, but think that it can be fun and perhaps one day the acquired skills on this subject could be put to good use. After all, at this level a supercomputer is not necessary: an affordable dual CPU system or two cheap, even obsolete PCs communicating via ethernet are enough for experimenting with a few parallel programming techniques.

The main possibilities

A detailed discussion of the theory of parallel machines is beyond the scope of this article. However, once the decision to write code for parallel computers has been made, it is wise to have at least a little knowledge of the platform that is going to be used. Not surprisingly, it turns out that there isn't only one type of parallel computer. Although at first sight the differences seem to be concerned with hardware, every architecture requires the use of different coding procedures for optimal performance. In this section we are going to give an outlook at the main implementations.

The most common parallel computers are called Multiple Instruction Multiple Data (MIMD): not only can they execute

A MIMD machine can execute the first **for** block on one processor and the second **for** block on another at the same time. This has to be contrasted with the example of a vectorial CPU like any modern consumer CPU, which can perform only different instances of a vectorial instruction. A Single Instruction Multiple Data (SIMD) computer is based on the same principle as the latter. The difference between a SIMD machine and a vectorial CPU is that parallelisation on a SIMD machine can be achieved by spreading the workload onto different CPUs. As a result, SIMD systems do not suffer from the problem of scalability of the number of instructions that can be executed at the same time. When compared with MIMD machines, SIMD machines are less flexible. Sometimes this reflects on performance. A common case is when there are conditional blocks like

```
if (i==1) {
/* do something here */
}
else {
/* do something else */
}
```

Suppose for instance that our computer has two processors, and in the first processor **i** is equal to **1**, while in the second **i=0**. In a SIMD machine, the second processor has to wait until the first has executed the first part of the block; the first processor will be idle in turn when the second processor executes the part associated with the **else** statement. In a MIMD computer, the second part can be executed in the second processor at the same time as the first processor is dealing with its bit of code.

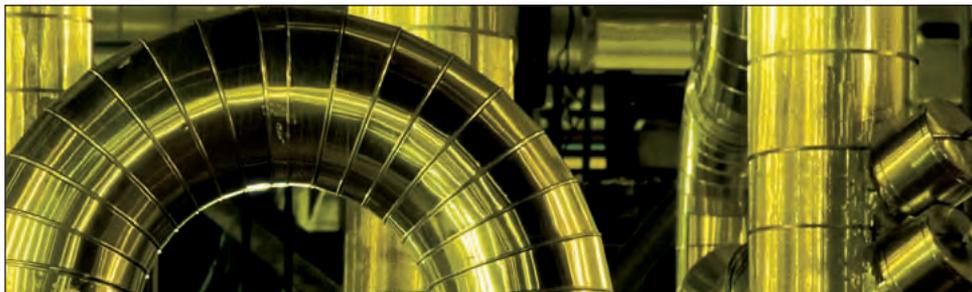
Before you dismiss SIMD machines as inefficient, you should be aware that they have also some advantages, an obvious one being synchronisation. To discuss it, let us complicate slightly one of the above examples:

```
for (i=0;j<100;j++){
a[i] = b[i] + c[i];
}
for (i=0;j<100;j++){
d[i] = e[i] + f[i];
}
for (i=0;j<100;j++){
a[i] += d[i];
```

“You need more processors as tomorrow’s weather forecast is no good if it arrives the following day”

independent vectorial operations like in the example above, but they can also execute non-interfering parts of the code containing different instructions. Consider for instance the following example:

```
for (i=0;j<100;j++){
a[i] = b[i] + c[i];
}
for (i=0;j<100;j++){
d[i] = e[i] - f[i];
}
```



While this example is not a template of good coding practice, and should not be taken too literally, these few lines suffice to discuss the issue. If on a MIMD machine it is chosen to execute the first and the second block on different processors, apart from the already discussed communication issue, at the beginning of the third block it is necessary that both processors have completed the assigned block. It would be a disaster if we attempted to add `d[i]` to `a[i]` before the value of the former has been computed. Whilst on a SIMD machine the workload must be distributed differently between the two CPUs, there won't be any problem of synchronisation: on a SIMD machine two processors can't do anything different from executing exactly the same instruction exactly at the same time.

The main reason why MIMD machines are more common than SIMD systems (apart from the price) is that they are more general-purpose. Among MIMD computers, it is usual to distinguish between shared memory and distributed memory systems. A shared memory system is a multi processor system where all the memory can be accessed by all the processors. A subcategory are the SMP systems, where one bank of memory is available to all processors. The main problem of this type of systems is scalability. Another problem is concurrent access to the memory, which degrades the performances. A generalisation of the SMP system that solves those problem is a system with different banks of memory available at the same time to all processors. This requires a more sophisticated technology, which translates to an increase of

price. The advantage of shared memory systems is that they are the parallel machines with programming paradigms closest to those of the serial machines. The disadvantages are related to synchronisation of variables and to mutually exclusive access to common memory areas. More on this later in this series.

On the opposite site we have shared memory machines, where each processor has its own reserved bank(s) of memory. The main disadvantage of this solution is that with the current library implementations details of communication are not transparent to the developer, although this can be also an advantage, since it offers better flexibility and possibility of choice. Still communication is the main issue on this type of machines, the main problems being related to possible idle times of processors while waiting for a message or the answer to a request they have submitted. There are library functions (Message Passing Interface, or MPI in short, being one of the standards in this field) that allow an efficient solution to this problem, but still the developer must be aware of code-related and hardware-related issues to make the appropriate choice of the communication method. The advantage of distributed memory systems is that in principle this solution can scale up to a very large number of processors in a cost-effective way.

Usually a hybrid solution able to combine advantages of both choices is adopted: the most common parallel computers (and among them many Beowulf clusters) have several distributed memory processing nodes, each processing node being a shared memory system with a few (usually two or four) processors. ■

NEXT MONTH

OUR FIRST LOOK AT the world of high performance computing terminates here. This quick overview has just scratched the surface. Here we have discussed when parallel programming is an appropriate choice and given a quick overview of different possibilities and related issues, but we haven't really written any parallel code and we haven't discussed how to port existing code. Next time we are going to dig deeply into parallel programming practice, writing some parallel code that (with the appropriate tools) runs on SMP systems. This will also clarify some issues related with coding in parallel for such systems.

Access

Richard Smedley investigates the latest efforts to make Linux comply with legislation and the needs of disabled users to interact with your business's computers.

Not enough companies go out of their way to make it easy for people with problems accessing computer technology to become employees. Current legislation in the USA, and recent bills soon to come into force in several countries throughout the world, have now levelled the playing field for disabled computer users – and I.T. departments who do not make all their firm's computers accessible are leaving themselves dangerously exposed.

There is a plethora of accessibility technology available to bolt onto your I.T. solutions: screen readers; voice synthesisers; braille terminals; and any number of input devices for those with limited mobility or motor control. While we touch on some of the assistive technologies this issue, we shall look in more detail at some of the Free Software and other solutions next month. For now we will examine why you should be adapting your I.T. infrastructure to make it more accessible, look at the issues that you should consider, and mention some of the solutions available.

Equality of opportunity

Even during an I.T. downturn, it is not always easy to get the best staff, and to hang onto them. Do you really want to limit yourself to a smaller pool of potential recruits, and ignore skilled and intelligent people who cannot use your computers until you make some (often small) changes to the hardware and software?

Of course simple human decency demands that we make our workspaces accessible to all potential employees, whatever their disability.

Unfortunately there are boardrooms where simple human decency fails to get a look in: for them there are persuasive arguments which can be made through the courts.

Most powerful of the recent bout of accessibility-related legislation making its way onto statute books around the world (see box, *Your company and the law*), is the USA's Americans with Disabilities Act (ADA), which, as well as affecting telecommunications and government agencies, applies to all American companies employing 15 or more people. Companies must take all reasonable steps to accommodate the disabilities of qualified applicants or employees, including modifying workstations and equipment. Remedy may be sought through private lawsuit, a disastrous result which companies are very keen to avoid.

Meanwhile in the UK, individuals and disability organisations are finding the Disability Discrimination Act 1995 increasingly relevant to workplace technology provision. If you want to avoid breaking the law, and to get the best choice of applicants, it's time to take a look at some accessibility solutions...

Graphical desktops

Use of a mouse requires visual ability *and* fine motor control. Keyboard navigation, then, is the cornerstone of accessibility – which will bring "I told you so"s from the command line *aficionado*. Unix, with its intimate association with text, has always had the potential to work well with screen-reading and braille terminal technology. However most of the world does not work the Unix way, particularly in the office.

The computer Graphical User Interface (GUI) that we know and love (or loathe) – the WIMP (Windows/Icons/Menus/Pointers) interface that was developed at Xerox PARC, licensed by Apple and mimicked by Microsoft



rights

“I.T. depts which do not make all their firm’s computers accessible are dangerously exposed.”

CHRISTMAS 2002

LINUXPRO



for their Windows OS – is not an ideal solution for everyone. In the words of Bill Haneman, GNOME accessibility hacker at Sun Microsystems, “the advent of GUIs was a real step back for blind users and a mixed blessing for those with severe mobility problems.”

Many a Unix admin will tell you that it is *always* quicker to work from the command line, and there are many blind and mobility impaired users who would agree. For blind and partially-sighted users the main issue is navigating the screen and reading the output. Working from the command line removes worries about window focus and menu navigation and concentrates on the important information without the resource-wasting gimmickry of the modern graphical desktop.

Output from the terminal is easily piped to assistive technologies such as screen readers and braille terminals. For example *Emacs*, not so much a text editor as a complete OS in its own right, integrates with the *Emacspeak* package, which talks to hardware or software speech synthesisers.

Linux newbies know *Emacs* as the incomprehensibly large modeless editor which is used by people who don't like *Vi*. There is a lot more to *Emacs* than this. *Emacs* is known

as the highly extensible text editor. It is easy to add functionality to *Emacs*, using lisp, and consequently hackers have added it by the bucket-load. Many Linux distros ship *Emacs* pre-built with a mail client, news reader and even a graphical web browser – *w3m-img*.

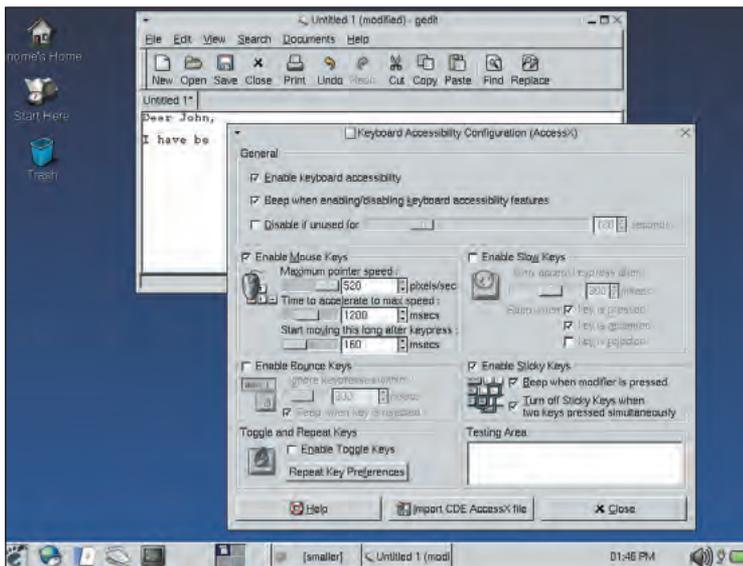
Emacspeak

Emacspeak provides fluent spoken access to local and remote electronic information through *Emacs*, enabling blind users to surf the Web, use instant messaging and write and manipulate documents through a well-integrated user interface. See the *Useful Links* box for full details of the lengthy list of supported apps.

Audio formatting is a technique earlier pioneered by *AsTeR*, giving items such as tables a contextual interpretation through the use of stereo. Full support for W3C's Aural CSS (ACSS) means *Emacspeak* opens up the Internet to sight-impaired computer users, without the high cost of a braille terminal. As *Emacspeak* can work with software (as well as hardware) speech synthesisers, there is now an even lower cost of entry.

With braille terminals costing US\$6-10,000, they are not the first solution to try and get

AccessX – a tool for movement-impaired users to customise the mouse, keyboard, and screen to their needs.



Here comes the Sun... Sponsoring GNOME accessibility

IT WAS BIG NEWS A COUPLE OF years ago when Sun announced that it was “throwing” 50 engineers at the GNOME project. After the fanfare died down the engineers continued at their coding. Only a few are working directly on the accessibility code - but what is important is that the project has had the goal of integrating GNOME accessibility right from the start, and they got started very early in the development cycle of GNOME2.

Sun’s contributions naturally also extend to accessibility improvements to Java, and to the accessibility of their own websites.

GNOME2.2 is now in pre-release, and scheduled for release in the first quarter of 2003. It will feature *complete* keyboard navigation as well as total theme compatibility (for low-vision users). It will expand the infrastructure for assistive technologies and should do much

more to make *Mozilla* accessible.

Sun’s work was recently recognised by the American Federation for the Blind (AFB), with the 2002 Helen Keller Achievement Award, for the Accessibility Framework for GNOME2.

When Pat Suelztz (executive vice-president of Sun Microsystems, Inc.) and Peter Korn (a GNOME accessibility engineer) accepted the award on behalf of Sun and the GNOME community, they spoke of the importance of open, collaborative accessibility solutions. Suelztz spoke of Unix as “not belonging to any single company, but is now in the hands of a larger community.” She also recognised the corporate contributions of Red Hat and Ximian, as well as the work of individual GNOME hackers, many employed by Sun. She then highlighted work on accessibility in *Mozilla* and *OpenOffice.org*.



Patricia Suelztz emphasised the community contribution to GNOME2’s Accessibility Framework.

past the bean counters. Hardware voice synthesisers are a lower cost solution and the most popular devices can be driven by all the common *nix screen-reading software. Nevertheless the price of this equipment will still be a bar to some - including voluntary organisations, but also those amongst the poorer countries of the world for whom technological access could mean a life-changing improvement.

Free speech

By the time Pentium class processors and motherboards with integrated sound cards became commonplace, software speech synthesisers became available to take advantage of this cheap commodity hardware, providing you and your company with a text-to-speech (TTS) solution, with no extra hardware needed.

ViaVoice, from IBM, is a popular non-Free cross-platform solution, which has been

bundled with some Linux distributions. However the most popular free software voice synthesiser is *Festival*, which can be made to work with most screen-reading software - including *Gnopernicus*, the GNOME2 screenreader. A Java TTS, *jsynth*, was developed for *freeplay*, by the Free Schooling

“The advent of GUIs was a real step back for blind users; a mixed blessing for the mobility-impaired”

project. It is available separately, to be incorporated into other apps.

So why bother with adapting GUI solutions, when we have *Emacspeak*? There are actually very good reasons for developing accessibility



FEATURE | ACCESSIBILITY



solutions around the same graphical desktops and applications used by everyone else in the office. Support is the primary issue here.

If a blind computer operator at, say, a call centre, has a problem with a buggy or misbehaving app, she can turn to a sighted co-worker and say “what’s going on on the

screen readers – such as the web browser *lynx* – are not so up-to-date when it comes to dealing with web technologies, even basic HTML tables. Webmasters have a duty to follow w3c guidelines on accessibility, but enabling everyone to use a standards-compliant browser such as *Mozilla* will ensure that everyone has equal access to the World Wide Web.

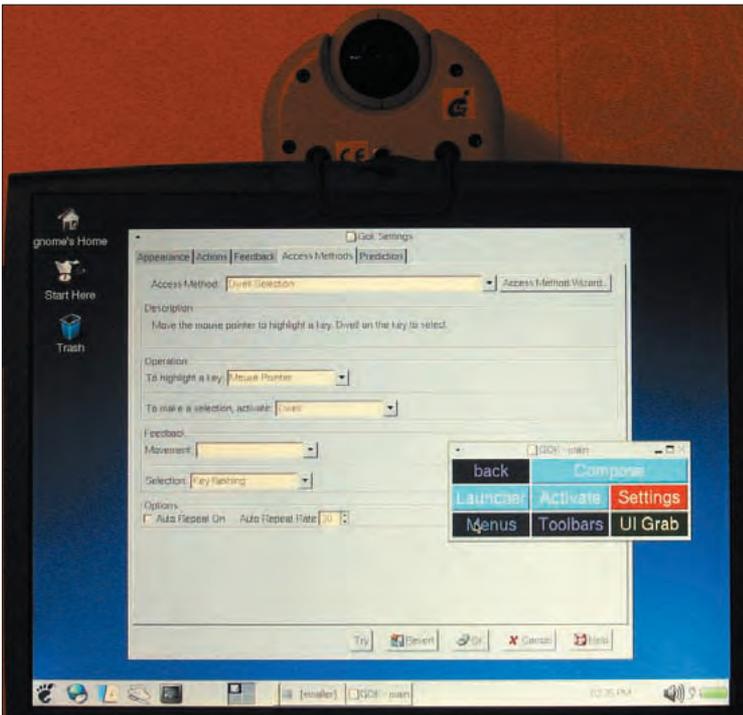
“There are good reasons for developing accessibility solutions around graphical desktops”

Assistive industry

It took eight years after the appearance of Microsoft’s Windows desktop for a screen reader to appear – but once assistive technologies were in place, Windows became known as the accessible OS. The bolting on of third party assistive technologies and accessibility solutions was the result of much reverse engineering and wrestling with obfuscated APIs. Current efforts to make the GNU/Linux and Unix desktop accessible, however, centre around

screen here?” – of course if they were running a command line app and the screen was filling up with debug information, there is little chance that their co-worker could help. Command line apps which work so well with

GOK – the GNOME Onscreen Keyboard – will work with any pointing device, including head pointer technology.



Your company and the law Keeping the lawyers happy

THE USA HAS THREE POWERFUL pieces of legislation:

“Section 508 of the Federal Rehabilitation Act requires that electronic and information technology developed, procured, maintained, or used by the Federal government be accessible to people with disabilities.” The Federal Government, through its agencies, employs *millions* of Americans.

“Section 255 of the Telecommunications Act provides that a manufacturer of telecommunications equipment or customer premises equipment shall ensure that the equipment is designed, developed, and fabricated to be accessible to and usable by individuals with disabilities, if readily achievable. Further, a provider of telecommunications services shall ensure that the service is accessible to and usable by individuals with disabilities, if readily achievable.”

“Americans with Disabilities Act (ADA) recognizes and protects the civil rights of people with disabilities and is modeled after earlier landmark laws

prohibiting discrimination on the basis of race and gender. The ADA covers a wide range of disability, from physical conditions affecting mobility, stamina, sight, hearing, and speech to conditions such as emotional illness and learning disorders.” It states that employers with 15 or more employees may not discriminate against qualified individuals with disabilities.

This latter legislation is enforced by the individual – and their lawyer – who may sue if employers do not comply. This is far more effective than leaving it to over-worked government inspectors, with little in the way of backup force, to ensure compliance.

Europa

Three years ago the European Commission launched the *eEurope* initiative, with the aim of “an Information Society for all.” Its objectives included making all of the EU’s public websites accessible to people with disabilities by the end of 2001 and establishing “centres of excellence” in each EU member

state to develop an EU curriculum in “Design-for-All” by 2002, so look out for one in your own country, if you live in Europe, where we seem to prefer bureaucrats to lawyers.

Legislation in Ireland, soon to come into force, should have the same effect as the American laws. The UK’s Disability Discrimination Act 1995 has recently been extended by the Special Educational Needs and Disability Act 2001. This act requires schools to ensure that all pupils have access to course materials – which nowadays also means to computers and computer programs. Other European legislation can be found linked from the GNOME Accessibility page.

Australia and Canada also have relevant acts on the statute books, but some of the most progressive legislation can be found in Shanghai. Here your company must employ a certain percentage of employees with disabilities. Failure to comply results in a large fine which is used to help projects employing disabled people.

integrating accessibility into the software.

Accessibility solutions can be either built-in, or added on assistive technologies. GNOME2 features advanced built-in capabilities – scheduled for further improvements in the soon-to-be-released GNOME2.2 – such as keyboard navigation, keyboard accessibility and theme compliance. Theme compliance is very important for those with low vision – enabling, for example, all applications to be displayed with high contrast or, for some users, *low* contrast.

High contrast themability is also very good for anyone attempting to use a laptop out-of-doors, in bright light. An example of accessibility efforts being of benefit to everyone.

Assistive technologies are add-ons providing additional features such as screen readers, magnifiers, and on-screen keyboards. GNOME2.2 gets the former two through *Gnopernicus*, and the latter with *GOK* – an onscreen keyboard, of which more next month.

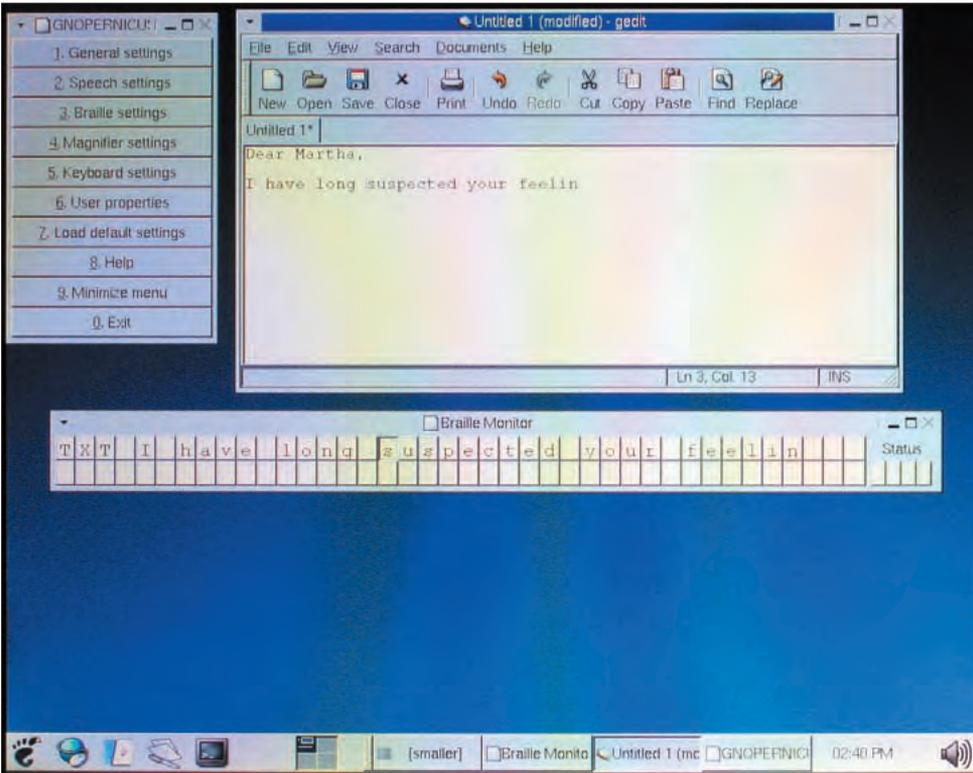
Positive result

It’s true that there are businesses (and individuals) who see the necessity of providing accessible workstations as yet another government-driven burden on business. Experience, however, shows that improving accessibility has benefits for all.

As well as improving software by cleaning-up APIs and making it more maintainable by



FEATURE | ACCESSIBILITY



Using the Gnopernicus screen reader to send text from a text editor to a braille terminal.



enforcing standard usage of widgets, the benefits of accessibility can spill into many areas. Wearable computers and various other embedded devices can benefit from technologies such as TTS and voice input.

More than this, many organisations are discovering the benefits of the disabled employees themselves. For example call centres have a notoriously high turnover of staff – but a much lower turnover amongst their blind and partially-sighted employees, who may be happier at the company, as they are working for someone that they know has made an effort that other employers sadly fail to do.

As well as encouraging employers – with carrot *and* stick – to provide technology in a way we can all access it, much more needs to be done to raise awareness amongst programmers and website creators.

Of course for coders (and those that pay their salaries) one could always appeal to their enlightened self interest – after all one does not know how easy one will find it accessing a computer in 50 days time, let alone 50 years – many of those who consider themselves “able-bodied” will one day benefit from assistive technologies and the accessibility features being developed for Free Software. ■

NEXT MONTH

Next month we take a closer look at some of the technologies – particularly the software – which make Linux such a good solution to your legal (and general) accessibility requirements.

Useful links A web of accessibility solutions

THE MAIN SOURCE OF INFORMATION should be the Linux Accessibility Resource Site (LARS) at:

www.tracecenter.org/linux and GNOME's accessibility pages at: <http://developer.gnome.org/projects/gap/>

along with the Free Desktop Accessibility Working Group (fdawg), which hosts a mailing list on accessibility:

www.speechinfo.org/fdawg
GNOME's own mailing list is very active:

<http://mail.gnome.org/mailman/listinfo/gnome-accessibility-list/>

as is its developers list: <http://mail.gnome.org/mailman/listinfo/gnome-accessibility-devel/>

Of course KDE are not neglecting the needs of their users, and as well as benefiting from the work of the GNOME project, are trying to improve KDE themselves with tools such as *KMouseTool*:

www.mousetool.com/ergonomic-software-for-linux.html and the screen magnifier, *KMagnifier*: <http://kmag.sf.net/>

The KDE accessibility mailing list: <http://mail.kde.org/mailman/listinfo/kde-accessibility/>

carries news and discussions on integrating accessibility software into future releases of KDE.

IBM maintains an Accessibility Centre at:

<http://www-3.ibm.com/able/>
As, of course, do Sun: www.sun.com/access

Legislation

Homepage of section 508 of the USA's 1998 Rehabilitation Act: www.section508.gov

More useful information on the

American 508 Act can be found at: www.section508.noaa.gov and many pages on UK, Australian and American legislation are linked from the CETIS-TechDis website:

<http://celt.bangor.ac.uk/accessibility/accessibilitylinks/legislation.htm> as well as from the GNOME accessibility site.

<http://developer.gnome.org/projects/gap/laws.html>

Blind + Linux

The BLinux (Blind + Linux) homepage is at:

<http://leb.net/blinux/>
Emacspeak:

<http://emacspeak.sourceforge.net/>
A full list of speech-enabled applications on the *Emacspeak* desktop:

<http://emacspeak.sourceforge.net/applications.html>

AsTeR (Audio System For Technical Readings) is a computing system for rendering technical documents in audio, developed by *Emacspeak* author T. V. Raman for his PhD in 1994:

www.cs.cornell.edu/home/raman/aster/demo.html

Project Ocularis:

<http://ocularis.sourceforge.net/>
the *Speakup* screen reader package has generated a whole community around it, encompassing a number of projects for blind computer users, which can be found at:

www.braille.uwo.ca/speakup

BRLTTY, a background process (daemon) which provides access to the Linux console for a blind person using a refreshable braille display, can be found at:

<http://dave.mielke.cc/brlitty/>
Many sites provide access to talking

books, including the education site, Engima Technologies: www.etc-edu.com/modules.php?name=BOOKS

HOWTO

Of course there's a HOWTO on Linux accessibility:

www.tldp.org/HOWTO/Accessibility-HOWTO/index.html

as well as one on how to develop accessibility solutions which was contributed by IBM:

www.tldp.org/HOWTO/Accessibility-Dev-HOWTO/index.html

Red Hat have a few basic tutorials linked from:

www.redhat.com/services/techsupport/accessibility

Jim Van Zandt's Linux *Emacspeak* HOWTO:

www.mv.com/ipusers/vanzandt/Emacspeak-HOWTO.html

And not forgetting...

Microsoft Accessibility: technology for everyone at:

www.microsoft.com/enable/ is quite informative, as is – Workability in Microsoft in the community (UK) at www.microsoft.com/UK/info/community/workability.htm

And a very useful resource for web designers, David Mertz's *Tips: Design for accessible Web sites: Making Web sites available to users with disabilities* – part of IBM's DeveloperWorks pages:

<http://www-106.ibm.com/developerworks/library/w-mertz.html>

W3C's own page on the Web Accessibility Initiative (WAI), contains guidelines and tips for making webpages accessible:

www.w3.org/WAI

FEATURE | BLENDER



ILLUSTRATION: STEFAN GARTNER – stefang@aon.at

FREE BLENDER

Nick Veitch interviews **Ton Roosendaal** and reveals the story of how the GPL has saved a valuable 3D tool.

18 **LINUXPRO**

March 2002, Linux 3D fans were dealt a terrible blow when the news emerged that NaN, the company behind the popular *Blender* 3D software, had filed for bankruptcy in the Dutch courts.

NaN were an unusual company in the 3D marketplace. Unlike heavyweight rivals behind products like *Lightwave* and *Maya*, NaN had always been distributed freely, although the code itself was closed and proprietary.

This fact, combined with the growing emergence of Linux as a key factor in the CG studios, made *Blender* very popular with amateurs and professionals alike. Advanced features, particularly the game engine, helped the software gain credibility.

Hit by the general IT recession, NaN found itself insolvent and unable to trade out of debt. The website was frozen, the code locked up and many fans around the world assumed that was the end of a rather great 3D tool.

Months past, but then in July, *Blender's*

CHRISTMAS 2002

original creator Ton Roosendaal announced he had made a deal with NaN investors, and for the sum of €100,000, they would license the *Blender* code to a new Blender Foundation, with the intent that it be released under the GNU GPL. Ton had 10% of the required money within a day, and about half had been pledged or received within a fortnight. The foundation was set up and finally, on 13th October, the *Blender* code was released.

Linux Format caught up with Ton, after a hectic time launching the first *Blender* conference, to ask him how 'Free Software' helped to save *Blender*.

LINUX FORMAT: What precipitated the crisis at NaN in the first place? Financial trouble?

TON ROSENDAAL: In companies, troubles usually have a financial aspect! Of course there are issues related to internal competencies, product quality and marketing strategy. Looking back, I see a brave attempt of a small Dutch company trying to make a difference in the 3D market worldwide. In some ways we were way ahead of the competition, but the market proved not to be ready for that. In fact, the market completely collapsed, something all 3D software companies cope with nowadays. Especially the prospectus for interactive 3D

and game prototyping tools didn't take off the way it was predicted early 2000.

Internally, we finished all targeted projects within the deadlines, but the revenues were way below planning. This caused the main investor in the company to pull the plug. In the investor's opinion, downsizing or a reorganisation was not a topic, we've had one chance to survive, and we didn't make it.

LXF: Did the investors need much convincing about releasing the code as GPL?

TON: Quite some! I proposed a survival strategy with a Foundation managing the IP by exploiting a GPL project months before the company closed. It took me three months of negotiating and putting pressure – behind the scenes – to get them to agree on the plan. The turning point happened when I proposed to collect a 100k fee from the community, as a feasibility study, to prove a Foundation strategy could potentially survive.

LXF: Do you think €100,000 was a fair price?

TON: Compared to what the investors lost, it is less than 10%. The amount was just a nice 'round' number. Still a substantial amount, something that would prove there was still belief out there in the product.

Get Blender!

THE SOURCE CODE may be downloaded from www.blender.org/modules.php?op=modload&name=Downloads&file=index

It will also be on next month's coverdisc.



ILLUSTRATION: STEFAN GARTNER – stefang@aon.at



The blender3d.org site is the public face of the new Blender Foundation.

FEATURE | BLENDER

At www.blender.org, a community site helps focus development efforts.



LXF: What made you so confident you could raise the money?

TON: Confident? Hah! I just like a challenge... the funding campaign was really a test. If it worked out well, the Foundation would have a clear future. Otherwise I just had to accept my baby would be locked up in oblivion forever.

Of course I had a strong motivation. There was not much choice here. "Pump or drown" as we say in Holland so nicely. Doing this, was also a promise I made to the user community three years before; I would accept investment

money to build a strong company, but when it fails I would do anything possible to make the sources open.

I started the NaN company in 1998, doing business by giving away the software itself for free. By running an e-shop I could manage to survive pretty well the first 2 years. At least that gave me confidence, that there's at least some business we can do with the current user base. Not to raise millions, but certainly to do a 100k within six months.

LXF: You might have been sure the money could be raised, but were you surprised by how quickly the target was reached?

TON: Yeah, happily surprised! I could only dream of such a result. I really was prepared for a six month campaign.

LXF: Where did most of the money come from? Were most of the contributors corporates or individuals?

TON: Individuals. I approached quite some companies, and most of them were positively interested. But the simple fact that we already collected $\frac{2}{3}$ of the money in about three weeks, made it quite difficult for them to understand that they were actually needed. Corporate sponsors require a clear plan what

Popular third party sites like elysiun.com will continue to feature Blender galleries and tutorials.



“Ton had 10% of the required money within a day and about half had been pledged or received within a fortnight”

you do with their money. I only had the campaign itself worked out, and not the actual Foundation activities once the sources were opened. For that I would have plenty of time. I thought!

LXF: Does the fact that *Blender* is now GPL mean any change in the direction of development?

TON: Hard to predict... I guess there will be quite some interesting forks available, which never should have been possible with a proprietary code project.

Within the Blender Foundation, using the blender.org platform, we will manage an 'official' release. Stable, tested and compatible releases. Especially in the beginning, the first months, we won't do radical things ourselves with *Blender*, but focus at bug fixing and feature requests instead. Setting up a decent project organisation will be far more important. Once we've reached that goal, serious work can be done to make a radical improved *Blender* version.

LXF: Is all of the *Blender* code now covered by the GPL?

TON: Unfortunately we didn't own the collision detection library in *Blender*. So in fact the current release is somewhat crippled, especially the game engine in *Blender*. Several people are working hard on a fix for this.

We also use the proprietary *FMOD* library for sound. It's a very good



CHRISTMAS 2002

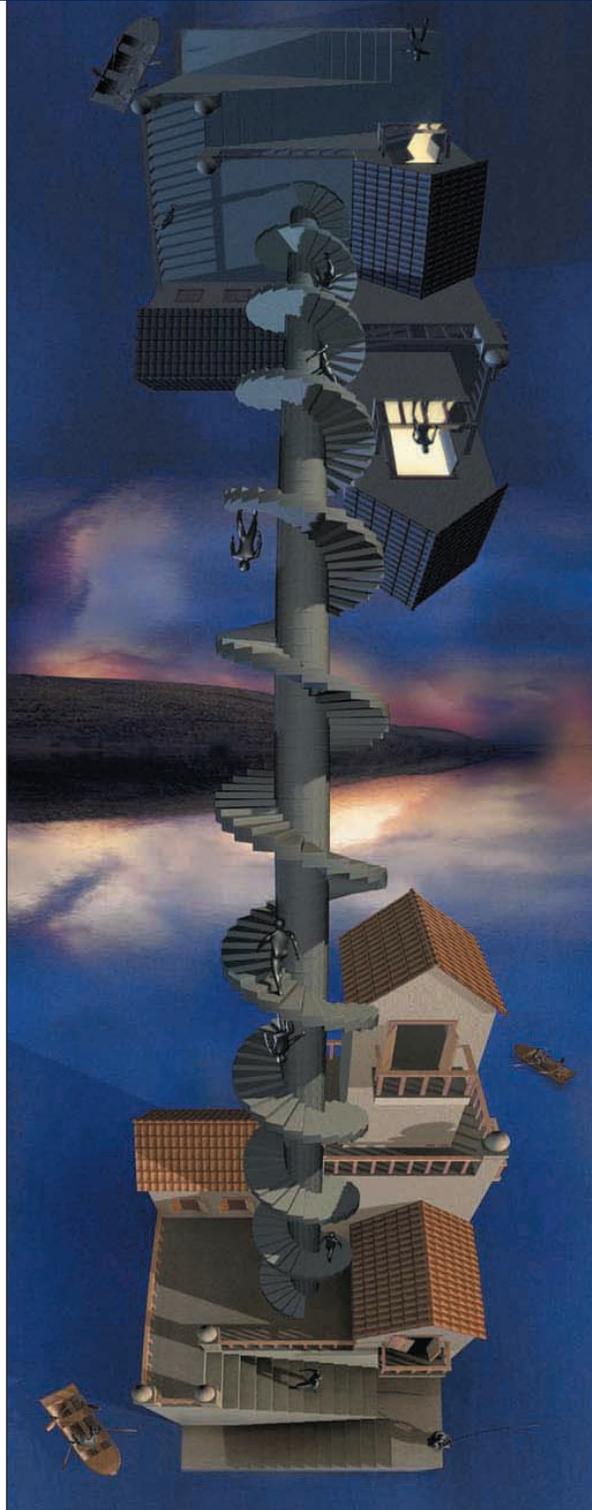


ILLUSTRATION: ANGEL QUIJADA ALVAREZ - ada_spain@yahoo.es

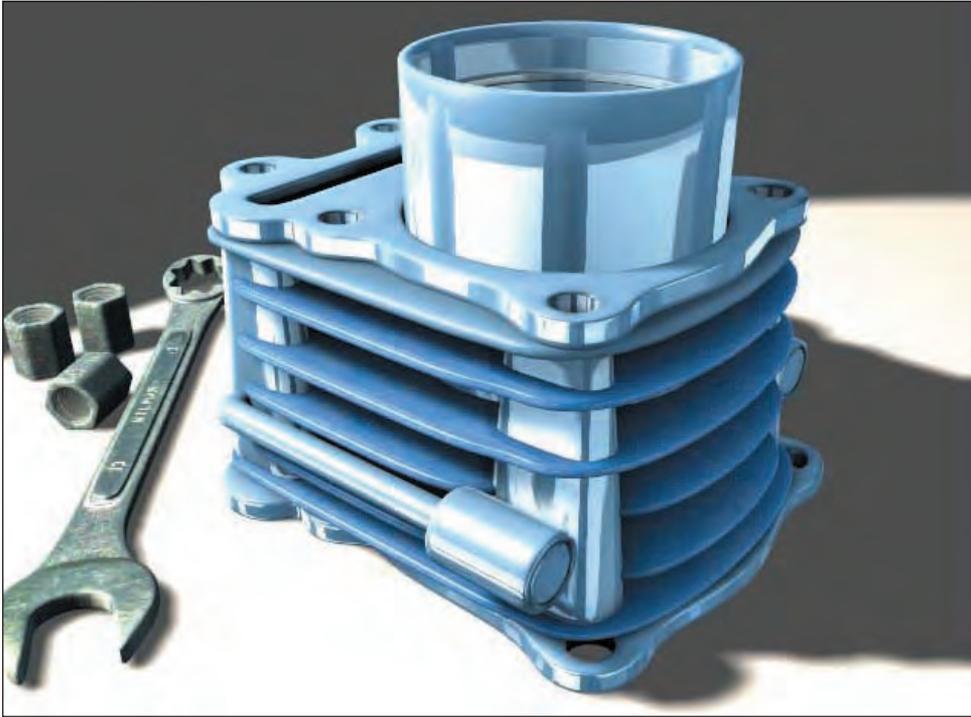


ILLUSTRATION: CHRIS PULSH – blengine@normal.com



sound system, but limited in licence... we have to evaluate that.

LXF: Do you think it's still possible to build a business around *Blender*, perhaps offering services, training or documentation?

TON: We decided to publish *Blender* under a dual license. Meaning that the original code is not only GNU GPL, but also available under a 'BL', a license that could be of interest for companies who – for whatever reason – cannot afford to incorporate GPL code in their products. Selling these licenses could be a nice source of additional revenues. And since the Blender Foundation is an independent non-profit public benefit corporation, these revenues will contribute to providing better services for everyone.

The plan is not to expand the Foundation to a big company. I prefer to keep the organisation as slim as possible, and facilitate a network of companies and professionals

around the Foundation who like to do commercial work with *Blender*. We are currently already in contact with a few companies about this. I would happily invite everyone to contact us, especially for setting up services and training. I don't consider that core business for the Foundation, but really hope others will pick that up in the near future.

LXF: Moving forward, what do you see as the key benefits of *Blender* now being free software?

TON: Of course a lot of people are very happy with the move. Especially the fact that this is the first Free Software 3D tool available, will help a lot of other GPL projects out there.

Whether we will be able to set up an efficient organisation to do functional releases is something only time will tell. *Blender* is not just a nice project for coders, there are thousands of artists out there who demand a product that works for them.

A very clear benefit that we already notice now, is that organising Free Software – within the non-profit sector – easily gets sponsors and volunteers for a lot of tasks and initiatives. In some ways, it even works more efficiently than having a large company doing it. Like doing the very successful *Blender* Conference last month. Our web services are also 100% sponsored now.

LXF: Have there been any disadvantages?

TON: Well, I've started something with an impact and complexity that is very hard to oversee...

LXF: Would it not have been worthwhile making *Blender* open source long before now?

TON: I don't know. In a way, I think it just happened at the right moment. The Internet hype is over, the landscape resembles a burnt-down forest... with very interesting new green sprouts popping up. I've jokingly mentioned we've entered "the dot org era", but seriously; it feels like being part of something bigger that's going to happen.

LXF: The new website has more of a community feel than the old one. I expect this is something you are keen to develop?

TON: The site blender.org will be devoted to development in general. For coders as well as for artists, we really need each other to get *Blender* developed. So there's a strong community aspect in it. The site is also far from ready.. we work at getting full project management running here, SourceForge style.

With a team of volunteers, we'll also build a 'product and usage' site at www.blender3d.org. Here we can reuse the entire old NaN database. That site won't have a community department, but will just provide the end-user services. We expect independent community sites, like the successful www.elysiun.com, to remain attractive. That's not something we should replace, but instead support where possible.

LXF: What are the immediate goals now that the money has been raised?

TON: Getting the services at the new sites up and running!

LXF: Will you continue to seek donations?

TON: Yep. Even with a slim organisation, a few people should be able to make a living of running the Foundation services. I expect more response from corporate sponsors now as well, and will apply for subsidy at EU and culture funds.

“I prefer to keep the organisation as slim as possible and facilitate a network around the Foundation”

LXF: When development gets moving again, what key areas of the software need work?

TON: That list is too long!

I prefer to call it 'bug & feature fixing'. That's priority number one, especially to keep *Blender* attractive for the huge amount of users out there. With the team who manages to do that, we then can sit together and discuss more radical changes.

LXF: We wish you luck, and no doubt, we'll be featuring your progress in future issues. ■

Blender offers many advanced features, and is particularly useful for game design.

