

72 SPEEDUPS

Is your distro feeling sluggish? Looking for that extra injection of pace?

Mike Saunders has spent the last six years performance-tuning Linux distros and still feels the need for speed... Additional writing: **Graham Morrison** and **Mark Bain**.

Question: what's better than a fast PC?
Answer: a *really* fast PC. Unfortunately, money constraints mean we can't always splash out on

hardware upgrades to make things faster. Fortunately, the software plays a crucial role too. In Linux's case, the modular arrangement of the kernel, startup scripts, GUI and apps means there's plenty to tweak for extra pace.

your box a lift. If you've been suffering in silence from a slow machine, or even if you've been raging loudly about sluggish startup times on forums the Linux world over, take some time out to try out some of these fixes. They could make a big difference.

We've researched and tested these tips on various distros, investigating ways to make them even more effective, and some are more demanding than others. Consequently we've given them difficulty ratings so

"THE KERNEL, GUI, APPS... THERE'S PLENTY TO TWEAK FOR EXTRA PACE."

In this bumper feature we've gathered together 72 of the most useful speedup tips, ranging from quickfire desktop hacks to more advanced memory-usage improvements. Bottlenecks and inefficiency can be found in many components of a modern desktop OS, so our tips cover bootup, desktop apps, servers and more. Whether you use Linux as a home desktop OS, development workstation or server, you'll find many of these hints give

you can tell how complex they are at a glance. We've also made a note of the software involved, if any. You'll find plenty of essential hints to try over the next 13 pages, so without further ado let's open the throttle and go...



EASY
MEDIUM
ADVANCED

01 Shorten bootloader timeout

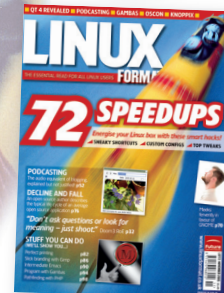
Software: *LILO/GRUB* • Difficulty: I

After the BIOS screen when you start your machine there's usually a pause while the bootloader offers you a choice of OS to start. If you're dual booting, for instance, you can select between Linux and Windows, or any other operating system you have installed. There's usually a period of time before a default entry is selected – ten or 20 seconds or so – but you can cut this down (making unattended boots significantly quicker).

If you're using the *GRUB* bootloader, edit *menu.lst* or *grub.conf*, which typically can be found in */boot/grub* or */etc/grub*, and then alter the **timeout** value to something short, such as three seconds. For *LILO*, edit */etc/lilo.conf* and modify the **timeout=** line, making sure to run */sbin/lilo* as root after to update it. Next time you boot, the pause will be much shorter.

DISCLAIMER

We've tested all these tips thoroughly, and the vast bulk of them are about software rather than hardware, but we accept no liability for any data loss or damage as a result of using them. *LXF* and Future cannot be held responsible if you toast your machine!



COVER
FEATURE

BOOTING

02 Stop services

Software: N/A • Difficulty: II



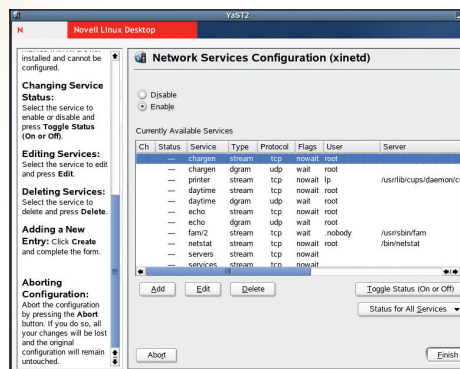
NICK SAYS...

"There's always the slight risk of disabling something that you really need. If you're in doubt, leave a service turned on for now, and read up on it. You can always turn it off later."

Between the kernel loading and login prompt coming up, various services (programs) are started, the number depending on which distro is running. You can usually see these services in the boot process with text saying Starting and then either OK or Done. Some services, such as the *hotplug* hardware detector, are best left alone; others can be removed to dramatically reduce boot time. For example, if you don't run a mail server, you can prevent *Sendmail/Postfix* from starting.

How to achieve this depends on your distro's startup scripts – they can vary wildly. Consult your distro's docs to find out where they live. On most systems, though, you can visit the */etc/init.d* directory and make any program you don't want started non-executable using **chmod -x** (or your file manager). Mandriva (*Control Center*), SUSE (*YaST*) and other distros include GUI apps to disable services. Have a look inside the program startup scripts for comments indicating what they do.

SUSE/Novell's YaST includes a mini utility for managing services.



BOOTING

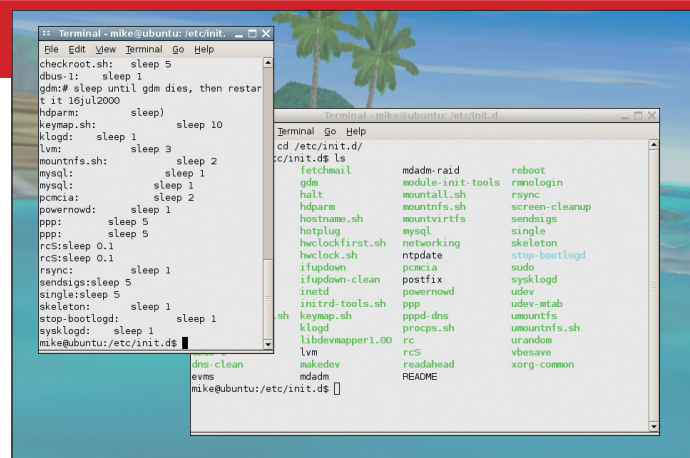
03 Cut down sleep(1) calls

Software: N/A • Difficulty: ■■■

If you poke around in your startup scripts, which are usually located in `/etc/init.d` in most distros, you'll find numerous 'sleep' references. The `sleep` command pauses execution by a specified number of seconds, and distro startup scripts often make heavy use of it when probing hardware or waiting for something to load. Naturally, this holds up other startup programs so it's worth finding out when it's applied. An example on a Debian-based system:

```
# grep -r sleep /etc/init.d
```

This will show which scripts contain `sleep` pauses, and display in seconds how long the pauses are (they're often several seconds long). You can experiment by removing some of these (or halving the length of time specified) for a faster boot, but be careful – if something is waiting for a hardware probe, things could get messy. Nevertheless, with some patience you can shave off five or ten seconds from your boot process.



On this Ubuntu box, quite a few of the scripts are sleepy...

BOOTING

04 Boot into text mode

Software: NXDM/GDM/KDM • Difficulty: ■■

When booting, starting the graphical login manager (usually `GDM` or `KDM`) can take a while. If you use mainly text-based apps, or you're running a server with some graphical programs installed, you can get your system running much earlier by booting straight into text mode. If you want to fire up the GUI later on, you can then run `gdm`, `kdm` (login managers) or plain old `startx` by hand.

The `/etc/inittab` file determines whether you boot to a GUI. This defines 'runlevels' for the system, ie which programs start (runlevel 0 halts the system, runlevel 1 is single user). Edit that file and find the line similar to `id:5: initdefault:`, then change the number in it. These numbers vary with distros, but on most you set it to 3 for text boot and 5 for GUI. Read through the rest of `/etc/inittab` for more info.

BEING ROOT

For many of the bootup, hardware and server tips you'll need to be logged in as root (the administrator user). Open a terminal and type `su`, and after entering the root password you'll be able to modify system files.



By logging in as root you can access and modify critical system files.

“IF YOU WANT THE LOGIN SCREEN TO APPEAR QUICKER, OPT FOR XDM.”

BOOTING

05 Reduce DHCP timeout

Software: DHCP • Difficulty: ■

Most modern PCs and laptops have some form of network card built in, and if Linux supports it, it'll most likely activate the card at boot time. If you've given it a static IP it should initialise very swiftly at boot – but if you get your IP via DHCP this can cause delays. It's especially true on laptops, where you often boot without a network cable plugged in, and the DHCP delay on boot can be 30 seconds upwards.

To shorten this, locate and edit the `dhclient.conf` file (usually in `/etc` or `/etc/dhcp3`). There's a `timeout` line where you can specify the number of seconds to wait. If your DHCP server responds quickly, change this to something in the region of five or ten. Next time you boot without a connection, it won't hang for as long as it did before.

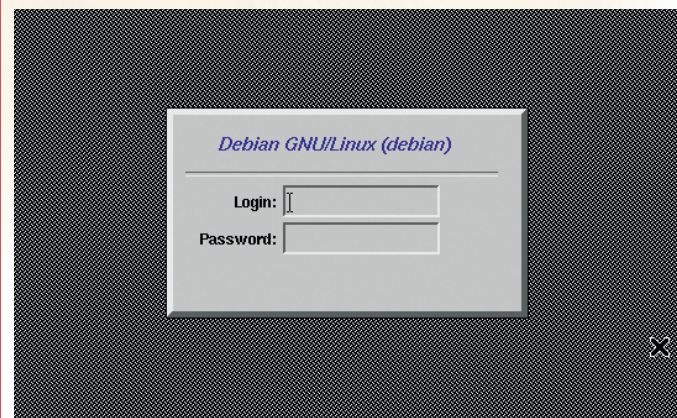
BOOTING

06 Quicker logins with XDM

Software: XDM/GDM/KDM • Difficulty: ■■

If you run Gnome or KDE, chances are you log in through their graphical display managers (`GDM` and `KDM` respectively). However, if you run a smaller window manager or just want the login screen to appear quicker, you can opt for trusty old `XDM` – the plain login manager supplied with `X`. It's rather unattractive in its default setup but pops up much quicker than the heavier Gnome and KDE equivalents.

To switch to `XDM`, you'll need to edit an `X` startup config file (which varies from distro to distro). Look in `/etc/X11/default-display-manager`, or `/etc/sysconfig/desktop` or `/etc/sysconfig/displaymanager`, change `kdm` or `gdm` to `xdm`, and reboot. Now the graphical login screen will appear more quickly.



It's ugly, it's bare, it makes babies cry – but `XDM` is fast.

BOOTING

07 Switch off graphical boot

Software: RHGB and other graphical boots • Difficulty: ■■

Some of the major distros sport nifty graphical bootup screens, complete with animations and other frills. (This is shown after the kernel loads but before the login screen appears.) It can add a few seconds to the boot – more for Red Hat, Fedora and CentOS – so if you don't need it, it's worth disabling. As with most tips involving startup scripts, there are many differences from distro to distro. In Red Hat and Fedora, remove **rhgb** from the kernel lines in **/boot/grub/grub.conf**. In SUSE, set **splash=0** and **vga=normal** in the bootloader config file. In Mandriva, you can disable it via the *Control Center* and remove the **splashimage** line in **/boot/grub/grub.conf**.

BOOTING

08 Start up lightning-fast

Software: *InitNG* • Difficulty: ■■■

Most distros start their boot time services and programs in sequence, which can waste time. Why not load a bunch of services when one is waiting for something else? *InitNG* loads services in parallel; that is, it doesn't let one service hold up the others, and boot times can be reduced by 50% or more.

The only downside is that *InitNG* doesn't work perfectly with all distros right now, and is still in heavy development. However, it's improving rapidly, so if you want to give it a try (and are willing to do some hacking at the command line), see our full HotPicks review on page 48 for more info.

BOOTING

09 Restart rapidly

Software: *Suspend 2* • Difficulty: ■■■

Many laptops support a hibernation feature, in which the contents of RAM are copied to the hard drive before shutdown, ready to reload without going through the entire boot procedure. Running programs are restored to the exact same state when the machine is next powered on. Software Suspend is a kernel patch that provides this capability for any machine. You'll need the kernel patch from www.suspend2.net along with the supporting *hibernate* script. You'll then need to rebuild and install your kernel, after which you can hibernate and restore at any point, skipping the full boot process.

Software Suspend 2 for Linux

Revolutionise the way you start your computer.

Software Suspend is most easily described as the Linux equivalent of Windows hibernate functionality. It saves the contents of memory to disk and powers down. When the computer is started up again, it reloads the contents and the user can continue from where they left off. No documents need to be reloaded or applications reopened and the process is much faster than a normal shutdown and start up.

Software Suspend 2.1.9.12 for Linux 2.6 is out!

Software Suspend 2 has a long [feature list](#), including the ability to cancel a suspend by pressing Escape, image compression to save time and space, a versatile plugin architecture, and support for machines with highmem, preemption and SMP.

The Software Suspend 2 website (this one) and [mailing list](#) provide support for dealing with issues arising. An IRC channel [#suspend2](#) has been created on [irc.freenode.net](#), where people may be able to help you with problems.

The primary author of Software Suspend 2 is Nigel Cunningham. A huge thanks must also go to Florent Chabaud, Pavel Machek, Gabor Kuti, Bernard Blackham and Michael Frank along with many others who have tested and contributed to the development of Software Suspend.

Current Versions

Here are the latest patches against both the 2.4 kernel series and the 2.6 kernel series. Note that patches for the 2.4 kernel series are no longer being actively developed and will only be updated for bugfixes and forward ports.

Latest stable version for 2.6 kernels: 2.1.9.5

- Software Suspend 2.1.9.5 for kernel 2.6.12

Latest stable version for 2.4 kernels: 2.1.5.7D

Transferring data from www.suspend2.net...

Suspend 2 is easy to install thanks to good documentation.

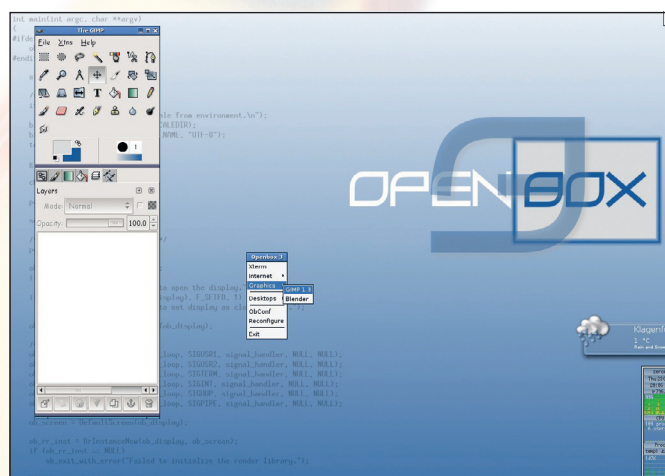
DESKTOP

10 Change Gnome window manager

Software: *Metacity*, *Openbox*, *Oroborus* • Difficulty: ■

Although Gnome's default window manager, *Metacity*, is described as a lightweight WM, there are various slimmer options available that use less CPU and RAM. One of the best is *Openbox* (<http://icculus.org/openbox>), which is on our coverdisc. Boasting support for chainable keybindings and customisable mouse actions in a diminutive package, *Openbox* is decently featureful and easy to use.

To replace *Metacity*, first compile and install *Openbox*. Then, in your Gnome session, run **openbox --replace** in the Run prompt (Alt+F2). This will switch the window manager. To make the change permanent, choose Save Session when you log out. For an even lighter replacement, try *Oroborus* at www.orooborus.org.



You can run Openbox outside of Gnome for ultra-minimalism.

DESKTOP

11 Launch KDE apps faster in other WMs

Software: KDE, various WMs • Difficulty: ■



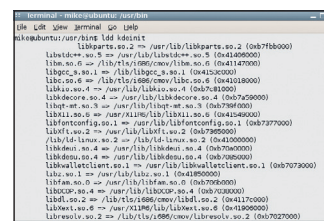
One popular option among all the speed-freaks out there is to run a lightweight window manager, with all your favourite apps running under it. However, if you run a lot of KDE programs under, say, *Fluxbox*, you'll notice lengthy startup times. This is because each time a KDE application starts outside its native desktop it has to load a stack of libraries and processes that support the app.

Fortunately there's an easy way round this. In your window manager, open up a terminal and enter **kdeinit**. This will load a lot of the KDE infrastructure, without changing your WM, and KDE programs will start much faster afterwards. You can put **kdeinit** in your WM's startup scripts to do this automatically.

PAUL SAYS...



"When you run *kdeinit* it starts up all the KDE background services, so you might as well just run KDE! Or *cough* Gnome..."



***kdeinit* links to numerous libraries - by having them loaded in memory, KDE apps start up faster.**

TIME-SAVING MYTHS

Myth: Compiling everything for specific CPUs is worthwhile.

Fact: A small handful of key system components, such as the kernel, *glibc* and other system libraries, can benefit from compilation with CPU-specific optimisations. However, rebuilding the likes of *OpenOffice.org* and *X-Chat* is

mostly pointless – you'll gain a tiny improvement in startup speed, but these apps spend most of their time waiting for user input, so the gains are negligible. Most desktop programs won't see any tangible benefit.

12 Enhance networked X performance

Software: X11, *lbxproxy* • Difficulty: I

If you're running X *Window System* programs over an Ethernet network, you won't find many performance issues. Over dial-up or other sluggish connections, though, it's not so pleasant, and is often so slow as to be unusable. Here's a really quick solution: the Low Bandwidth X (LBX) Proxy. Programs on your machine don't need to know about the LBX extension but they connect to this proxy, which performs optimisations on the X protocol. And it's absolutely seamless.

lbxproxy is supplied with many X.org installations or provided as a separate add-on. Install it and read through the manual page (**man lbxproxy**) for more information on how it works. It's also a good idea to scan the options to see which optimisation and compression techniques can be disabled if they cause problems with a certain app.

13 Consider using FVWM

Software: *FVWM* • Difficulty: II



KDE and Gnome provide excellent desktops and give you everything you could possibly want. However, this comes at a cost in the amount of time that the desktop takes to load, and in the amount of memory used. If you have an older PC (or even if you have a new one but you want to optimise the performance), then switching to a lighter window manager will minimise the amount of memory used on your PC, and even reduce the time that it takes to log on. If you use *FVWM* you'll also find it incredibly easy to customise, giving you the essentials at your fingertips. See www.fvwm.org.



www.xwinman.org details almost every window manager in existence.

DESKTOP

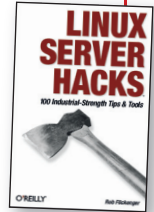
14 Boost app startup by 50%

Software: *prelink* • Difficulty: III



One of the problems with software applications is that they depend on other software to run. When an application accesses a library, special symbols are transferred from the library to the memory that the application is running in. Unfortunately, this copying process (called linking) can take quite some time.

On most systems, libraries are rarely changed, and when a program is run, the process of transferring these symbols is the same every time. A special tool called *prelink* uses this repetition to link once and store the result of the process in a file that can be executed. This method can greatly improve performance, particularly in C++ software. Some KDE software runs up to 50% faster. *prelink* is on our coverdisc, and a full guide can be found at www.craet.us/james/articles/prelink.php.



15 Launch Gnome apps faster outside the desktop

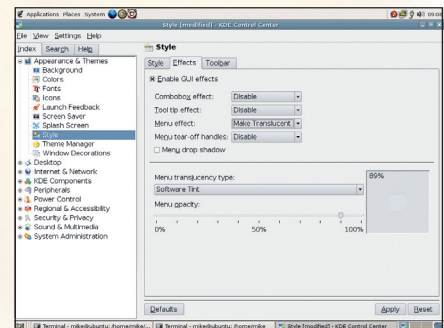
Software: Gnome, various WMs • Difficulty: I

This follows a similar principle to Speedup #11. If you're running a lightweight window manager such as *Window Maker*, Gnome apps can take an age to start, because they load a bulging heap of supporting Gnome infrastructure (such as the comically over-engineered *gconfd*). But there's a good way to mix some of the top-notch Gnome programs with a slimline WM: just fire up a small Gnome app at the start of your session – *gedit* or *gcalctool*, say – and minimise it without closing. Other Gnome apps will start significantly faster from there onwards, and even if *gedit/gcalctool* gets swapped out it's still quicker than loading from scratch.

16 Tune KDE's visual frills

Software: KDE, *kcontrol* • Difficulty: I

KDE includes a staggering array of cosmetic touches, many of which can be disabled for a healthy stride forwards in performance. Your first port of call should be the *KDE Control Centre*, and the Style settings panel under Appearance & Themes. By switching to the Effects tab you can turn off animated or translucent menus among other CPU-munching features. Disabling the splash screen yields a slight improvement in startup speed, while getting rid of anti-aliased fonts and a complicated background image also helps in terms of processor and RAM usage. Also check out the Translucency options under the Window Behaviour entry (found in the Desktop settings list) in the KDE Control Centre.



You can turn off all visual frivolities for a plainer but snappier desktop.

DESKTOP

17 Choose an old-school distro

Software: N/A • Difficulty: ■

If you have an old PC (such as a Pentium 1), try not to be tempted by those lovely, brand-spanking new distros on the cover of the latest *LXF*. They look pretty and you can try loading them but you will find that your PC will become almost unusable. No, you need to be looking at your collection of back issues. Find your disk with Slackware 10.0 on it. Hunt through your draws and grab that copy of SUSE 7.1. Yes, you may want the newest version, but frankly you might not have the memory or the processing power required. Another answer is, of course, to get down the shops and buy yourself a new PC. But isn't that the easy way out?

“ANOTHER ANSWER IS TO BUY A NEW PC. BUT ISN'T THAT THE EASY WAY OUT?”

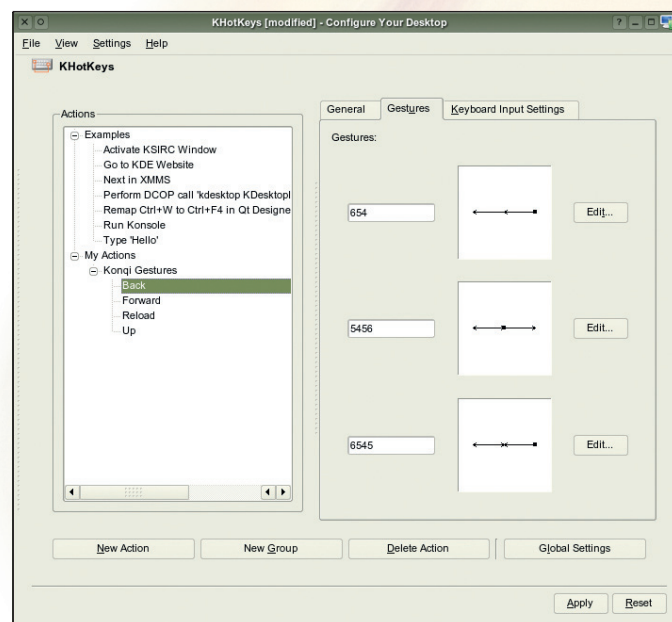
DESKTOP

18 Use KDE mouse gestures

Software: KDE • Difficulty: ■

A mouse gesture is a simple swipe of the mouse made while pressing a certain mouse button. It could be as simple as moving your mouse from left to right. You can also construct your own more complicated gestures, such as a circle or a saw tooth. Gestures are used commonly to move between pages in a web browser, but with KDE you can navigate your filesystem or control certain programs using *Konqueror*.

Gestures can replace any key command. They are created and managed from the *KDE Control Centre* under the module labelled KHotkeys. They first need to be turned on from the General Settings tab, followed by creating your own gestures in the My Actions list in the main window (the default ones never seem to work).



Neatly, lines and arrows show a visual representation of the gestures.

DESKTOP

19 Make Nautilus snappier

Software: *Nautilus* • Difficulty: ■

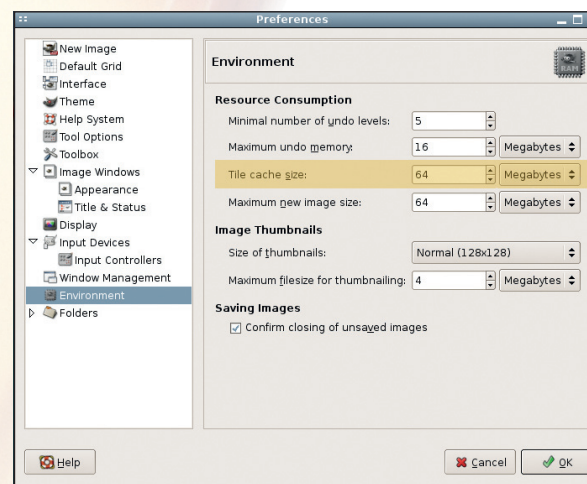
Nautilus, Gnome's file manager, has been on an impressive diet for the past few years, morphing from a wobbling mass of code to a respectable, mildly demanding program. You can still speed up its overall operation by altering a few settings in the Prefs dialog (found under the Edit menu). In the Preview tab, set as many options as you like to Never – the file browsing windows won't be as lively, but the benefit is that *Nautilus* won't try to read through every previewable file it encounters. The defaults are to avoid extensive reads on network files, but it also makes a difference locally.

APPS

20 Improve Gimp tile cache usage

Software: *Gimp* • Difficulty: ■

Gimp uses its own mini swap system – in other words, it moves memory contents to the hard disk when they've not been accessed for a while, or resources are low. The Tile Cache setting specifies how much RAM it should use before temporarily storing work on the hard drive, and is usually set at 64MB by default. Go to File > Preferences > Environment to tweak this. If you regularly edit large images without running many other apps at the same time, set this at around half your RAM for a simple but effective speedup.



Tweaking the tile cache size is easy via the Prefs dialog.

APPS

21 Preload Konqueror

Software: *Konqueror* • Difficulty: ■

KDE's file and web browser can be preloaded in the background so that new instances are loaded almost immediately. This is a tactic copied from Microsoft Windows, and it is effective because a typical KDE session loads *Konqueror* many times. This setting is found in *Konqueror*'s Performance page, accessible from the Configuration window. You can choose the number of instances you wish to be preloaded and a single instance that must always be available. The default number is three, which is a sensible choice, but it depends on your typical working routine. You may need to increase or decrease this number but the more you use, the less free memory you will have available on network files; it makes a difference locally too.

APPS

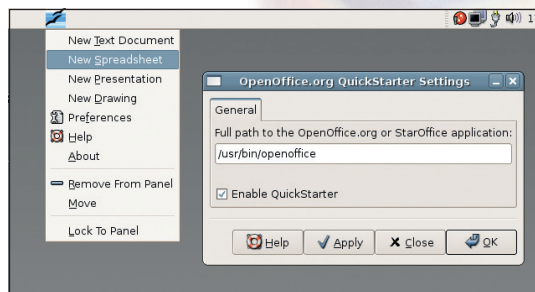
22 Start OpenOffice.org quicker

Software: *OpenOffice.org* • Difficulty: ■



ON DISC

Amazingly powerful as it is, *OpenOffice.org* still crawls along at startup. Some efforts have been made to speed up version 2.0's loading process (see page 70 for more on that!), but not with any amazing results yet. Pleasingly, then, a couple of projects provide short-term fixes for this problem, by keeping the core *OpenOffice.org* components permanently in memory. So when you start *Writer* or *Calc*, most of the office suite is already loaded in memory, and windows pop up in two seconds rather than 20. It's not a perfect solution but does improve productivity immensely. There are two quickstart tools available, both on our disc: for Gnome, <http://ooqstart.sourceforge.net>; or KDE, <http://segfaultskde.berlios.de/index>.



The Gnome OOo QuickStarter is merely a panel applet that sits quietly in the background.

TIME-SAVING MYTHS

Myth: Many distros compile specifically for ancient 386 PCs.

Fact: This idea stems from two things: the fact that most RPM and Deb packages have the numbers '386' at the end, and the fact that they're not built with a high -march GCC flag. In reality, most distros build packages using the

386 instruction set for compatibility with many processors (Intel, AMD, Via and so on) but with these instructions highly tuned for more modern chips (-mcpu flag). It's not as wasteful as it initially appears.

APPS

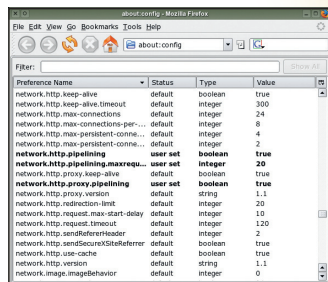
23 Browse faster with Firefox

Software: *Mozilla Firefox* • Difficulty: ■■

Firefox is fast, but if you're using a broadband connection you can make it even faster. The trick is to increase the number of requests it can make at any one time. This is the web browsing equivalent of multi-processing, and it means that the browser can download other things while waiting for any other requests to be filled. To enable this feature, enter **about:config** into *Firefox*'s location bar; this should then provide a long list of parameters. To change a parameter's value, double click on its line. **network.http.pipelining** and **network.http.proxy.pipelining** need to be set to True, and **network.http.pipelining.maxrequests** needs a value (20 to 30 range to works well).

GRAHAM SAYS...

"Beware that some web servers don't support pipelining. If you get errors, turn this setting off and re-enable afterwards."



Config items you've changed are highlighted in bold.

SUPERFAST TIP



APPS

24 Switch to IMAP

Software: Mail clients • Difficulty: ■■

The POP protocol has been in use for years, and it's what most people use to download their email. Most internet providers now offer an alternative in the form of IMAP. IMAP has some distinct advantages over POP – it doesn't download email, which makes the whole process much, much faster, and it also means you can access your email from anywhere on the internet.

Both *Evolution* and *KMail* support IMAP folders, and as long as your internet provider supports the protocol, it's just a matter of changing the server. If you subscribe to a great number of mailing lists, using IMAP avoids not only the downloading of the messages, but also the processor overhead when sorting messages.

APPS

25 Eliminate browser stalls

Software: All web browsers • Difficulty: ■

Every once in a while you visit a web page that seems to take forever to load, or simply hangs and never finishes loading. The problem often occurs because the web page includes a picture, button or advertisement from another server that your browser cannot access quickly.

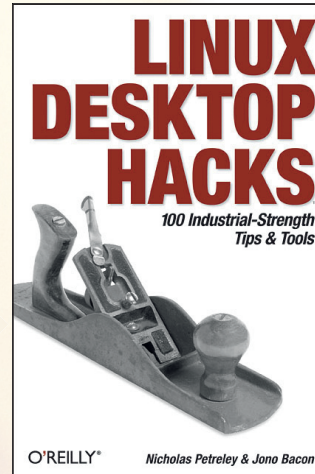
You can short-circuit this process so that it never has to contact that problematic external website (ie the one serving the adverts or images). If your browser stalls with 'Looking up <sitename>' or 'Waiting for <sitename>'

and that site is external to the one you're visiting, edit your **/etc/hosts** file and add a line that defines the problematic web site so that it points to the IP address for your localhost (your own machine):

127.0.0.1 www.someadvertisementsite.com

The main site should load quickly now – when your browser tries to access the advert from the external site, it'll receive nothing and move on.

One of the authors of *Linux Desktop Hacks* is sometime LXF scribbler Jono Bacon. The man knows his stuff!



APPS

26 Give Metacity some lick

Software: *Metacity* • Difficulty: ■

By default, *Metacity* (Gnome's window manager) includes a rather ugly minimise animation together with a few other graphical effects than can be disabled for a quick speed boost. You have to do this via *gconf-editor* – this is usually found in the Applications menu under System Tools, and is called Configuration Editor. Go to / > Apps > Metacity > General, then check the Reduced_resources box.

The blocky minimise animation will disappear, as will opaque resize and move (good CPU savers for older machines). To revert to the previous behaviour, just uncheck that box in *gconf-editor* – the changes are applied straight away.

APPS

27 Renice your processes

Software: *All apps* • Difficulty: ■

Sometimes it's unfair for one program to hog all the CPU, especially if it's something you want running quietly in the background while you do other work. There's a way to change the priority of processes, which can speed up your most-used programs considerably: *nice*. By running **nice -n <number> <program>** from a shell prompt you can give that program priority; the number is from -20 (highest) to 19 (lowest). Only root can start programs with negative (higher) priorities. Similarly, you can change the priority of a running program with **renice**. See **man nice** and **man renice** for more info, and **top** to view current process priorities.

APPS

28 Save time with Firefox and Konqueror searches

Software: *Mozilla Firefox, Konqueror* • Difficulty: ■

You can save considerable time by using the quick search functions of both of these browsers. A quick search is entered into the same field you'd type a site's URL into. In *Konqueror*, for example, typing **gg:linux format** will search Google for your favourite magazine. If you replace **gg** with **ggi** you will search Google images instead. Other worthwhile prefixes for *Konqueror* include **dict** to search the Merriam-Webster online dictionary, **thes** to search the thesaurus, and **wp** to search Wikipedia. You can add your own search destination from the configuration window. *Firefox* works in the same way, except it uses a slightly different syntax. To search Google, just enter **g linux format** into the location bar.

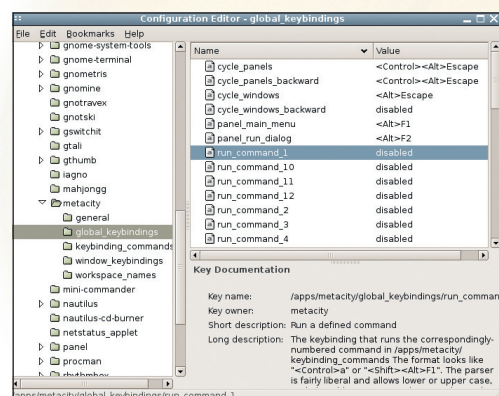
The keyword, or the letter you use before the search, can be changed by right-clicking on the Bookmark to the site and selecting properties.

APPS

29 Create keybindings in Gnome

Software: *gconf-editor* • Difficulty: ■

You can create keybindings to start your favourite apps in Gnome without trekking through the menu. Launch *gconf-editor* (usually found under System Tools in the Applications menu), and navigate to / > Apps > Metacity > Global_keybindings. Along with the default keys for window operations there are several numbered **run_command** entries. Add your keybinding to one of those, switch to **keybinding_commands** in the left-hand tree and insert the actual command (matching the number) on the right. You can, for instance, map **Ctrl+Alt+F** to start up *Firefox*.



Click where it says 'disabled' and you can tap in a new keybinding.

30 Get networked in an instant

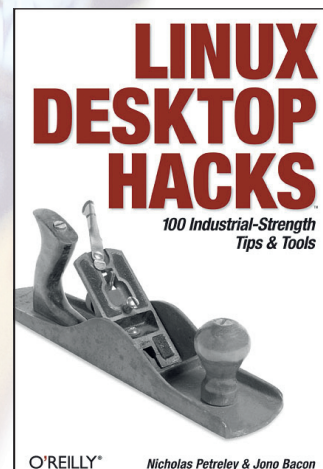
Software: *ifplugd* • Difficulty: ■



ON the DISC

ifplugd (on our disc) is a daemon that watches your Ethernet connection to see if it is live or disconnected. Plug the wire into the network and *ifplugd* configures the interface, optionally with DHCP. Unplug the wire, and *ifplugd* disables the interface. It's a perfect utility for laptops and workstations that frequently change their network connections.

The *ifplugd* utility simply checks your network interface(s) to see if they have a link beat, which indicates a live connection to a network. When a link beat appears, *ifplugd* configures the interface as being up and ready to use. When the link beat disappears (you disconnect the cable), *ifplugd* brings it down. See the docs in the source or <http://Opointer.de/lennart/projects/ifplugd> for usage details.



31 Shorten OpenOffice.org's startup

Software: *OpenOffice.org* • Difficulty: ■

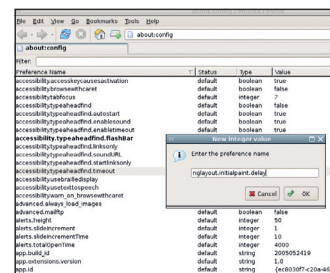
If Speedup #22 doesn't float your boat, there's another way to reduce *OpenOffice.org*'s alarmingly slow startup times. Launch one of the programs (*Writer*, for example), open the Tools menu and select Options (at the bottom). Beneath the *OpenOffice.org* menu there's a sub-entry for Memory. In Graphics Cache, change the Use For *OpenOffice.org* number to 30MB and the Memory Per Object to 2.0MB. Subsequent launches should be noticeably faster – reports from users vary, but many have found it to be highly effective. Combine with #22 to make startup actually tolerable!

32 Make Firefox render quicker

Software: *Mozilla Firefox* • Difficulty: ■

Normally, *Firefox* pauses slightly when loading a page before it begins drawing it to the screen. This is to make sure most of the structural content is loaded – otherwise the display would shift and morph rapidly as data comes down the line, creating a rather disconcerting effect. Conversely, it does waste a little time, and you can speed this up by setting a new option. In the URL bar, enter **about:config**. We need to add a new preference name by right-clicking in the list, selecting New and Integer and entering 'initialpaint.delay'. This also needs to be given a value of 0. Now pages will render slightly faster.

The about:config screen looks a mess at first glance, but you can filter through the list.



APPS

33 Inject pace into EclipseSoftware: *Eclipse* • Difficulty: ■

There are a handful of things you can do to improve the performance of the *Eclipse* IDE. On older machines, these are godsend – it's a hefty beast at times. You can give the JVM more memory to play with (ideal if you're not running anything else) by altering the Java command line. Try **-Xms96m -Xmx256m** to specify minimum and maximum memory of 96 and 256MB respectively. Disabling the Automatic Code Folding and Automatic Code Insight features can also give your machine an octane boost – go into Preferences from the Window menu, choose Java and Editor, then uncheck Enable Auto Activation along with everything in the Folding tab.

DATABASE

34 Tune PostgreSQL shared buffersSoftware: *PostgreSQL* • Difficulty: ■■■

Shared buffers are a block of memory used to hold requests until the kernel can respond and manage the data. By default, this setting is quite low and can be enlarged for better performance – but if it's too high it can have the opposite effect. The best setting depends on your machine: 4MB (512 shared buffers) is appropriate for a development workstation; 16 to 32MB (2048 to 4096) works well on a box with 256 to 512MB of RAM using a medium size data set; and for the high end (a system with 1GB+ of RAM), it's best set at somewhere between 64 and 256MB (8192 to 32768).

TIME-SAVING MYTHS

Myth: Speed and stability are separate.

Fact: Some GCC optimisations, specifically CFLAGS, can produce crash-prone code. GCC's manual page explains which options are safe and which can produce unstable binaries. As some Gentoo users have seen when compiling a whole system with extreme optimisations, many programs react

badly to over-zealous tweaking. The **-fomit-frame-pointer** flag can turn some apps into a catastrophic crash waiting to happen. So when making optimised source builds, it's safest to stick with **-O2 -march=<chip>** (or possibly **-O3**). This gives a slight boost without dangerously over-optimising volatile

DATABASE

35 Avoid MySQL lower and upper statementsSoftware: *MySQL* • Difficulty: ■

In *MySQL*, avoid using the lower and upper statements in a select statement – they slow the query down and you don't need them. The *MySQL* select statement is not case-sensitive: **select * from customers where surname='jones'** will return exactly the same result as **select * from customers where surname='JONES'**. Surprisingly, **upper(surname)='JONES'** is treated exactly the same as **lower(surname)='JONES'**. Also, avoid using the wildcard (*) in a select statement (such as **select * from mytable**). This appears useful but will cause the query to run more slowly, and creates extra network traffic. Instead, only call for the fields that you actually require, such as **select email from customers where surname='jones'**.

DATABASE

36 Get faster queries with indexesSoftware: *General databases* • Difficulty: ■■

SUPERFAST TIP



NICK SAYS...

"Optimising databases isn't the most exciting work, but a combination of these tips can lead to an immediately noticeable boost."

Don't forget to add indexes to the tables in your databases. A correctly placed index can make the difference between a database query taking a fraction of a second and tens of minutes. Choose fields that you expect queries to be run on. For instance, on a table of customers you are most likely to make queries on surnames and first names so use **create index ind_cust_name on customer(surname, firstname)**, depending on the names of your tables and fields of course. You should also take note of any fields that you are using in joint statements, as the tables may be fine by themselves but together they can be slow.

DATABASE

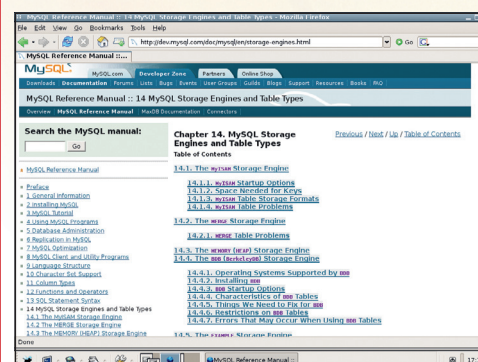
37 Disable fsync for disk boostSoftware: *PostgreSQL* • Difficulty: ■

In many *PostgreSQL* installations, the **fsync** option is enabled, causing the server to write every operation to disk immediately. This helps to maintain the database's integrity during an OS crash or hardware failure, but it's also slow. Disabling **fsync** leaves the disk-write operations up to the OS, which can then buffer and order writes in the most efficient way – delivering a large performance leap. If your hardware and power supply are reliable, you can disable **fsync** in *PostgreSQL*'s config file. Tread carefully if the server's flaky, or your power source is dodgy and you don't have a UPS!

DATABASE

38 Select the right storage engineSoftware: *MySQL* • Difficulty: ■■

MySQL offers a variety of engines for storing data, so it's important to select the best one for your particular task. *MyISAM* is the default on many setups, offering superb performance but without the integrity guarantees of transactions. *InnoDB* is slightly slower (around 20% slower in many general benchmarks) but includes the aforementioned transaction support. The *Memory* (formerly known as *Heap*) engine provides extreme performance – it's not a reliable choice, though. *CSV* and *Archive* should be avoided unless you have very specific requirements. On the whole, *MyISAM* and *InnoDB* are the best choices for maintaining speed.



See *MySQL*'s documentation for more details on storage engines.

SERVER

39 Build PHP for speed

Software: *PHP* • Difficulty: ■

Although most distributions provide binary packages for PHP, you can gain a few speed boosts by building from source. The **CFLAGS** settings can add a few percentage points of pace, but it's the **./configure** options that inject a lot more. Most crucially, make sure to use **--disable-debug** to chop out the debugging code, which you won't need on a production box, and also use **--enable-inline-optimization**, which allows PHP to generate optimisations in the code. Lastly, for PHP 4.2.0 and newer, **--enable-mm=shared** is a good idea (fast shared memory for session storage).

```
--with-all-opts      include new XSL support (requires libxslt >= 1.0.0)
--enable-yp          use the libyast install directory.
--enable-yp          include YP support.

--with-mysql          Install MySQL in RPM (default: /usr/lib/mysql)
--without-mysql      Do not install MySQL

--enable-maintainer-cs enable thread safety - for code maintainers only
--enable-inline-optimization If building and_execute fails, try this switch.
--enable-memory-limit   Compile with memory limit support.
--enable-cond-multiple Compile with cond multiple support.

--with-force-opts-config Use GNU PHP.
--with-force-opts      Use POSIX threads (default)
--with-force-opts      Use POSIX threads (default)

--enable-shared[=SO]    build shared libraries (default)
--enable-static[=NO]    build static libraries (default)
--enable-fast-install[=NO] optimize for fast installation (default)
--enable-ipv6           compile the C compiler with IPv6 (default)
--enable-libxml          enable libxml support (default)
--enable-openssl         try to use only OpenSSL objects (default)
--enable-sockets         try to use only POSIX sockets (default)
--enable-zlib            try to use only ZLIB objects (default)
```

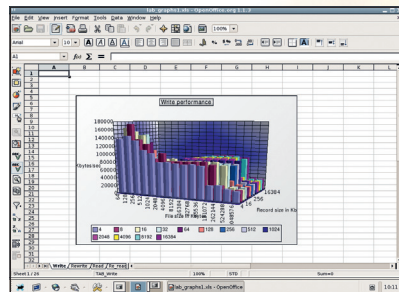
Run **./configure --help** to get an exhaustive list of PHP build options.

SERVER

40 ReiserFS for the mail queue

Software: *Various MTAs* • Difficulty: ■■

If you're running a mail server that's processing thousands of messages a day, it's worthwhile choosing the right filesystem for its queue. This is typically stored in **/var**, so formatting **/var** with a different filesystem type can yield a boost. Some Linux FSs don't run too speedily with thousands of small files in a single directory – consequently, it's worth changing from ext2/3 (the default with many distros) to ReiserFS. This filesystem has proven to be zippier when handling a large number of small files, and helpfully, most major distros support it in their default kernel so it's not a massive chore to switch. On high-load servers you'll see a good improvement.



Benchmarking tools such as **IOzone** (www.iozone.org) are useful for ascertaining the performance of your filesystem.

SERVER

41 Craft the right balance between client and server

Software: *N/A* • Difficulty: ■

If you're building a client-server application, consider balancing your load between the server and the client. If all the work is done on the server it could slow the whole application down, even though the client software is doing nothing. If all the work is done by the client this can also slow the application down. Look at using stored procedures on databases, or whether to read data from the database and then process it in the client. Think about this when you're developing the software, because at first you won't see any problems. It's only when you've got lots of users that the service will start to noticeably degrade – and by then it may be too late.

SERVER

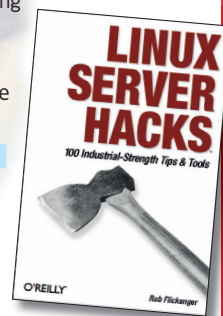
42 Keep CPUs busy with parallel builds

Software: *make* • Difficulty: ■

If you're running a multiprocessor system (SMP) with a moderate amount of RAM (say, 512MB), you can usually see significant benefits by performing a parallel **make** when building code. Compared with doing serial builds when running **make** (as is the default), a parallel build is a vast improvement. To tell **make** to allow more than one child at a time while building, use the **-j** switch:

```
# make -j4; make -j4 modules
```

Some projects aren't designed to handle parallel builds and can get confused if parts of the project are built before their parent dependencies have completed. If you run into build errors, it is safest to just start from scratch this time without the **-j** switch.



“COMPARED WITH SERIAL BUILDS, A PARALLEL BUILD IS A VAST IMPROVEMENT.”

SERVER

43 Improve Postfix failed message handling

Software: *Postfix* • Difficulty: ■

When the **Postfix** mail server can't deliver a message, that message sits in the queue, wasting space and time as the server periodically tries to resend it. If you have a lot of mails that can't be delivered, this can blunt performance, so it's worth tweaking some parameters. **maximal_backoff_time** (in seconds) can be set high so that **Postfix** doesn't keep retrying too often. **queue_run_delay** (also in seconds) specifies how often the server scans for deferred mail to resend – again, worth setting higher if you're overloaded. The **maximal_queue_lifetime** (days) parameter can be knocked down a few notches to avoid problematic mails hogging the queue and server processes for too long.

SERVER

44 Purge .htaccess files

Software: *Apache* • Difficulty: ■

Don't use **Apache** **.htaccess** files. In fact you should stop **Apache** from looking for them at all. There is nothing that these files do that can't be done in the **httpd.conf** file, and looking for them just adds load to the server. For instance, if your **htdocs** directory is **/var/www/htdocs** then **Apache** will look for **.htaccess**, **/var/.htaccess**, **/var/www/.htaccess** and **/var/www/htdocs/.htaccess** – four unnecessary file accesses for every web page request whether or not the files exist. To disable **.htaccess** files completely, set the **AllowOverride** parameter in your **httpd.conf** file to 'None':

```
<Directory />
    AllowOverride None
</Directory>
```

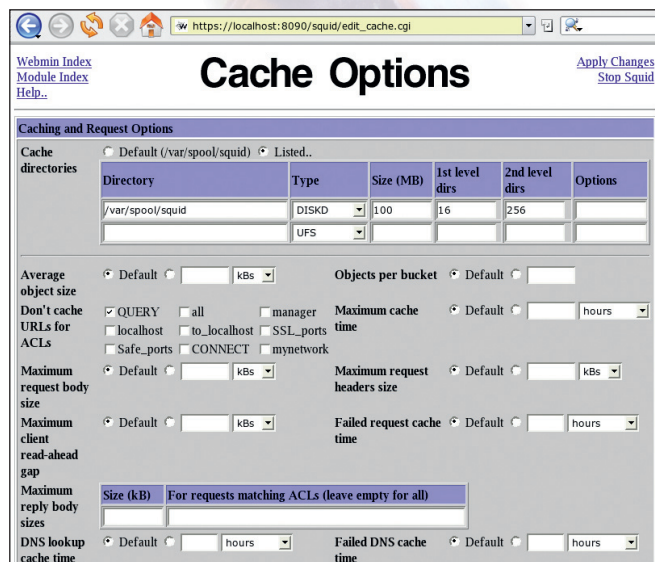
SERVER

45 Cache websites

Software: *Squid* • Difficulty: ■■

ON the DISC

If you run a network of machines, you can vastly improve the access speed by using a proxy server like *Squid* to cache the more popular websites. *Squid* is a proxy server that acts as a gateway to the internet, but it also saves the most common sites locally so it doesn't need to trawl the internet every time someone requests the latest BBC headlines. Managing the *Squid* server itself is best done through a web interface like *Webmin*, which makes changing the various settings, like the size of the cache, easy.



One of the sanest routes to *Squid* configuration is via *Webmin*.

SERVER

46 Use async with NFS

Software: *NFS* • Difficulty: ■■

The network file system (NFS) is quite an efficient way of accessing remote data over a network as if it were stored locally, but without the **async** option it suffers from terrible performance. Without **async**, data is written before each write request is completed, whereas with the option enabled, the server can write the data when it's most convenient to do so. This has obvious speed advantages, but if the server crashes before it performs the write, the data will be lost. To enable **async** transfers, it needs to be added to each entry in the `/etc/exports` file. A typical example would look like this:

```
/mnt/media 192.168.1.0/255.255.255.0(rw,no_root_squash,async)
```

SERVER

47 Select ext3 journalling type

Software: *ext3*, *mount* • Difficulty: ■■■

Most distros that offer a journalling filesystem during installation default to ext3, the mature evolution of Linux's standard ext2 filesystem. The type of journal can affect performance significantly: with the **data=journal** mount option, all data is committed to the journal before writing – the safest but slowest. **data=ordered** is the default, but you can try **data=writeback** for extra speed (and the risk of finding old data in files after a crash).

SERVER

48 Disable Apache modules

Software: *Apache* • Difficulty: ■■

Remove any *Apache* modules you don't need. Many of the modules are very useful (if not completely vital). However, there are many that are never used but are still loaded, and therefore use additional memory that could be used elsewhere. Fortunately it is easy to experiment with most modules by editing your `httpd.conf` file. Don't delete any lines – simply use a hash at the start of a line to remove the `AddModule` and `LoadModule` statements for the particular module. You can then use `apachectl restart` to restart *Apache* and see what effect there is. It's time-consuming, but worth doing. This is of course only suitable for modules that are not directly linked to *Apache* – these need a complete rebuild.

KERNEL

49 Rebuild the kernel for your CPU

Software: *Linux kernel* • Difficulty: ■■

SUPERFAST TIP

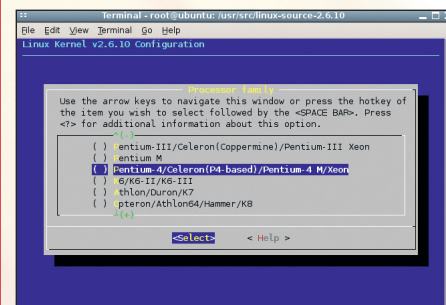
It's well worth having a kernel tuned for your processor. This, together with an optimised *glibc*, makes a system snappier than the generic 386 support. Recompiling the kernel can be lengthy, but it's not difficult if you follow a good guide. You'll have to obtain the kernel source for your

distro (or a standard release from <http://kernel.org>) and then build it. A quick Google search will find a kernel compilation guide for your distro; or you can ask on the *LXF* forums. When you're at the configuration phase, drop into the Processor Type And Features section where you'll need to select your CPU type from the Processor Family menu, then recompile, install and reboot.

PAUL SAYS...



"Many distros provide their own kernel packages optimised for specific CPUs. Have a look before you compile it by hand."



The Linux kernel has optimisations for lots of CPU types.

KERNEL

50 Rebuild the kernel to remove features

Software: *Linux kernel* • Difficulty: ■■■

Most distros provide a large, all-encompassing kernel that's designed for a variety of users and hardware. If you know exactly what you need, though, you can trim down the kernel to save a small amount of memory and CPU cycles. Having only the features and hardware support you need is great – determining what you need is the complex part. Get the kernel source as described in the previous Speedup, then in the config menus step through each option (consulting the online help if necessary) and chop out anything you definitely don't require. You'll find you can slice off whole subsystems (such as SCSI), but if you're in doubt about something, leave it in.

KERNEL

51 Fine-tune your kernel with sysctl

Software: *sysctl* • Difficulty: ■■■

Although most system settings can be tweaked via **/etc**, there are some options that require a bit more technical nous. The **sysctl** tool (enter **man 8 sysctl** for a brief guide) modifies certain kernel settings while it's running. These settings often have bizarre names, yet can be useful for improving performance on production machines.

Enter **sysctl -a | less** to view all available options. Even with the jumbled names, you can identify what most of them do (for example, networking settings have a **net.** prefix). The hundreds of options are beyond the scope of this tip, so to find out more, search the web for **sysctl** and the aspect you

```

File Edit View Terminal Go Help
dev.rtc.max-user-freq = 64
dev.parpport.parpport.devices.lp.timeslice = 200
dev.parpport.parpport.devices.active = none
dev.parpport.parpport.devices = PCSPP,TRISTATE
dev.parpport.parpport.dma = -1
dev.parpport.parpport.iq = 7
dev.parpport.parpport.base-addr = 956 0
dev.parpport.parpport.spintime = 500
dev.parpport.default.spintime = 500
dev.parpport.default.timeslice = 200
dev.cdrom.check_media = 0
dev.cdrom.lock = 1
dev.cdrom.debug = 0
dev.cdrom.autoclose = 1
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
dev.cdrom.info = drive name: hdc
dev.cdrom.info = drive speed: 61
dev.cdrom.info = drive # of slots: 1
dev.cdrom.info = Can close tray: 1
dev.cdrom.info = Can open tray: 1
dev.cdrom.info = Can lock tray: 1
dev.cdrom.info = Can change speed: 1
dev.cdrom.info = Can select disk: 0
dev.cdrom.info = Can read multisession: 1

```

want to improve. **Sysctl** plays a heavy part in low-level network tuning, so that's one area worth investigating.

The lines may look like gobbledygook initially, but they hold the key to some low-level performance tweaks.

KERNEL

54 Accelerate QEMU

Software: *QEMU* • Difficulty: ■■

QEMU, a full PC emulator, is great for testing distros. Until recently, every CPU instruction had to be emulated, and it wasn't majorly fast as a result. Now there's a kernel module that effectively turns **QEMU** into a virtual machine à la **VMware**, and it's much zippier.

Download the **kqemu** kernel module from <http://fabrice.bellard.free.fr/qemu>, and build it (you'll need your kernel sources) according to the documentation. The next time you run **QEMU**, it'll latch on to the module for a large performance gain. Definitely worth the slight hassle of installing your kernel sources.

TIME-SAVING MYTHS

Myth: Disabling services saves lots of CPU and RAM.

Fact: In Speedup #2, we showed how to prevent certain services from starting at boot time, getting the system up and running much quicker. However, it doesn't mean your system will be faster or much lighter in use – most services sit idle and get swapped out (virtual

memory), so they rarely hog any memory or processor. If you never use **Sendmail**, for instance, but leave it starting on boot, it'll sit quietly and be swapped out when other apps run. The real benefit to disabling services is boot-time improvements.

KERNEL

52 Make better use of cache and swap

Software: *sysctl* • Difficulty: ■

For this tip, you should read #51 first to learn about **sysctl**. Linux 2.6 introduced a **vm.swappiness** **sysctl** setting, which defines how virtual memory is used. Normally, when a program needs more memory and RAM is already full, the kernel can either: a) reduce the disk cache, which stores files in RAM for quick access; or b) move some running programs on to the disk (swap space) for temporary storage.

This **sysctl** setting, between 0 and 100, tells the kernel which method to favour. At 0, the kernel will discard most of the cache when memory is low, whereas at 100, programs will be swapped out quickly. For best responsiveness, set it to 20 or 30 (**sysctl -w vm.swappiness=20**). It's worth experimenting with other values and storing the best in **/etc/sysctl.conf** to save between reboots.

HARDWARE

55 Get better hard drive throughput

Software: *hdparm* • Difficulty: ■■



ON the DISC

DMA (Direct Memory Access) allows data to be transferred

without heavy load on the CPU. Most newcomer-friendly distros enable DMA on hard drives, but some don't, and performance suffers greatly as a result. Enabling DMA can improve disk performance by five times or more.

Install **hdparm** and run as root: **hdparm /dev/hda**. Replace **hda** with **hdb** or whatever your drive is named. This will show you what features are enabled; to turn on DMA, use **hdparm -d1 /dev/hda**. You can use the **-i** switch to show what features the drive supports, and **man hdparm** gives more info. Other performance switches worth trying on your hard drive (and CD/DVD) include **-c** for 32-bit I/O support and **-X** for IDE transfer mode.

SUPERFAST TIP

GRAHAM SAYS...

"Caution! Some **hdparm** flags have been known to corrupt data. The man page highlights the troublesome options."



KERNEL

53 Add an ultra-responsive kernel patch

Software: *Linux kernel* • Difficulty: ■■■



ON the DISC

The Linux kernel is an all-round performer for servers, desktops and other machines. Consequently, various patches are doing the rounds which focus specifically on making desktops highly responsive. The most famous is Con Kolivas's patch bundle. This features a different scheduler (staircase), less file cache usage and tuned HZ value. Kolivas makes patches for all the latest kernel releases – so just download, apply (**man patch**) and rebuild your kernel to get a snappier, smoother desktop. See <http://members.optusnet.com.au/ckolivas/kernel> for the patches and www.linux-militia.net for pre-patched kernel sources. There's an RPM for Fedora on our coverdisc.

```

File Edit View Terminal Go Help
root@ubuntu:~# hdparm -i /dev/hda

/dev/hda:
Model=HTS424030M9AT00, FwRev=MAA7A75A, SerialNo=MPAA32QBHS4LB
Config=( HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs )
RawCHS=16383/15/63, TrkSize=0, SectSize=0, ECCbytes=4
BuffType=DualPortCache, BuffSize=1739KB, MaxMultSect=16, MultSect=16
OurCHS=16383/15/63, CurSects=14481935, LBA=yes, LBASects=58605120
IORDY=on/off, tPIO=(min:240,w:IORDY:120), tDMA=(min:120,rec:120)
PIO modes: pio0 pio1 pio2 pio3 pio4
DMA modes: mdma0 mdma1 mdma2
UDMA modes: udma0 udma1 udma2 udma3 udma4 *udma5
AdvancedPM=yes, mode=0x80 (128) WriteCache=enabled
Drive conforms to: ATA/ATAPI-6 T13 1410D revision 3a:

* signifies the current active mode

root@ubuntu:~# hdparm -t /dev/hda

/dev/hda:
Timing buffered disk reads: 76 MB in 3.04 seconds = 24.98 MB/sec
root@ubuntu:~#

```

Use **hdparm -t <device>** to see what difference the tweaks make.

HARDWARE

56 Set your mouse and keyboard pace

Software: Any window manager • Difficulty: ■

For many people, a fast keyboard repeat rate and accelerated mouse are essential for moving around large documents. Some desktops and window managers include their own config tools for mouse and keyboard settings, but if yours doesn't, you can fall back to the standard `xset` tool supplied with all X packages. Here's an example of tweaking the keyboard repeat rate:

```
# xset r rate 500 30
```

This sets 500 milliseconds from a key being pressed to the repeat starting. It then repeats 30 times a second.

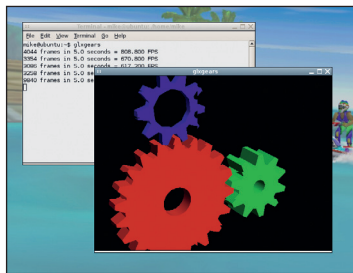
“BOOST SPEED BY USING PROPRIETARY VIDEO DRIVERS FROM NVIDIA.”

HARDWARE

57 Enable 3D support in X

Software: X.org • Difficulty: ■■

If you're using a distro with a heavy desktop focus, such as Mandriva or SUSE, it's most likely that the installer correctly configured the X GUI for you first time around. However, if you're doing things by hand you may need to configure 3D support. This isn't too tricky providing you know what you're doing. First, make sure your kernel supports AGP (enter `dmesg | grep -i agp` and `lsmod | grep -i agp` in a terminal to see if there's an `agpgart` line). If not, you'll need to rebuild the kernel with the MTRR, AGP and DRM options enabled. Add **Load "dri"** and **Load "glx"** lines to the Module section of `xorg.conf` (usually found in `/etc/X11`) and restart X. Entering `glxinfo | grep direct` should then result in 'Yes'.



Test your 3D performance with `glxgears` – it periodically notes the frame rate.

HARDWARE

58 Play with vendor-made video drivers

Software: N/A • Difficulty: ■■

The standard X.org video drivers supply adequate to good performance for most video chips, but in some cases you can boost speed by using proprietary drivers straight from the vendor. NVIDIA's drivers, found at www.nvidia.com/object/unix.html, are split into kernel and X.org modules, so you'll need your kernel source installed to build them. After installation, edit your `xorg.conf` to replace the `nv` Device line to `nvidia`, and remove the `dri` Module Load line. ATI's drivers are at www.ati.com – the docs supplied explain how to install. In both cases you have to sacrifice some freedom as in speech for a speed jump, but for gaming it can be well worth it.

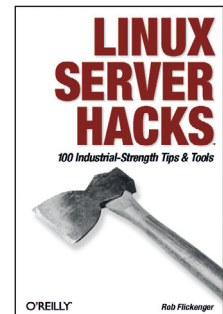
HARDWARE

59 Use every byte of your RAM

Software: Linux kernel • Difficulty: ■■

Linux is capable of addressing up to 64GB of physical RAM on x86 systems. But if you want to accommodate more than 960MB of RAM, you'll have to let the system know about it. First of all, your Linux kernel must be configured to support the additional RAM. The way that the kernel addresses its available system memory is dictated by the High Memory Support setting (aka the `CONFIG_NOHIGHMEM` define setting.) Set this according to the amount of RAM you intend to use.

Be warned that selecting 64GB requires a processor capable of using Intel Physical Address Extension (PAE) mode. Once the kernel is built and installed, you may have to tell your boot loader how much RAM is installed, so it can inform the kernel at boot time (as not every BIOS is accurate in reporting the total system RAM at boot). Add `mem=<number>M` to your kernel line in the `LILO` or `GRUB` config files.



HARDWARE

60 Try the mouse in text mode

Software: GPM • Difficulty: ■



If you haven't got the X Window System installed on a machine, it doesn't mean you have to live without the mouse entirely. There's a spiffy little tool called `GPM` (*General Purpose Mouse*) which provides a simple mouse system for the CLI: a block cursor, together with copy and paste facilities. This can be immensely handy in, for example, an IRC client or mailer that doesn't include a fully-fledged editor (nor let you access one). `GPM` can be found at <http://linux.schottelius.org/gpm> and is available in most distros' package archives – but you may get flamed by CLI die-hards!

HARDWARE

61 Overclock your NVIDIA card

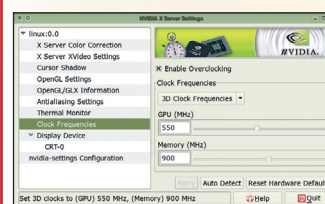
Software: nvidia-settings • Difficulty: ■■

Overclocking is dangerous. You can permanently damage your hardware, so please proceed with caution. Recent NVIDIA drivers include the ability to run your graphics card's processor and memory at a faster rate than they were designed for. A small increment can increase performance, but it can also cause your card to overheat. To get this to work, you need the latest driver from NVIDIA's homepage. Edit the X configuration file, usually found at `/etc/X11/xorg.conf`, and add the following line to the NVIDIA device section:

```
Option "Coolbits" "1"
```

After restarting the X server, you can change the clock speed from the `nvidia-settings` application included with the driver. With Coolbits enabled, there's an extra page labelled Clock Frequencies where you can adjust the

speed of the GPU and the memory. Proceed with extreme caution and make only small changes to avoid system meltdown.



No need for any command-line faffing around, it's all done through this slick GUI.

HARDWARE

62 Cut down sound lag

Software: *ESD, Artsd* • Difficulty: ■

You've got the latest and greatest soundcard, you've installed the newest, snazziest distro and you're raring to go. But for some reason, you're getting delays in game sound effects and when playing music. Often, this is caused by the sound daemons used in Gnome and KDE, called *ESD* and *Artsd* respectively. These daemons act as middlemen between the hardware and software, allowing several apps to connect to them and play sounds simultaneously, but there can be a second or two's lag because they're not going straight to the hardware via the usual `/dev devices/`. You can avoid this by disabling the daemons (kill them with a process manager if necessary) and setting your programs to direct *OSS* or *ALSA* output.

CLI

63 Compress SSH connections

Software: *OpenSSH* • Difficulty: ■

If you use the *SSH* remote login tool over a slow connection, you'll often find a high latency between typing and the characters appearing. *SSH*ing between two boxes on dialup lines often results in horrible juddering. Thankfully there's an easy way round this: compression (sending less data down the line). By using the `-C` flag with *OpenSSH* you can apply compression to the connection, improving response greatly, and this uses the same methods as *gzip*. You can even apply `-C` to *scp*; however, if the file or files you're transferring are already compressed it'll only slow it down. Here's an example login command with the compression flag:

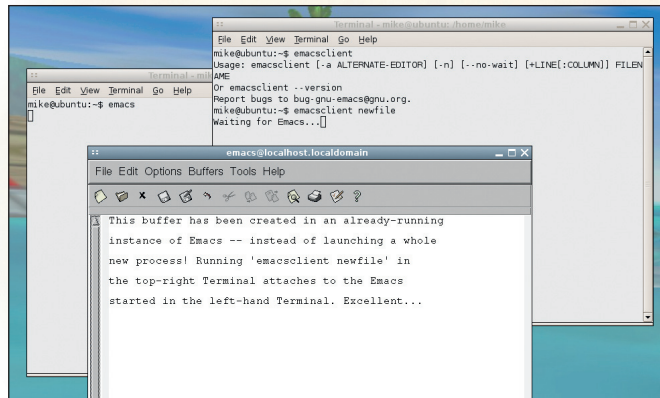
```
# ssh -C user@hostname
```

CLI

64 Open Emacs instantly

Software: *Emacs* • Difficulty: ■

Most modern systems have no problem loading *Emacs*, but it was once mockingly given the expanded name 'Eight Megs And Constantly Swapping' because of its memory usage. On old machines it can take a while to start, but there's a useful solution in the form of *emacsclient*. Add a `(server-start)` line to your `~/.emacs` file, and run *Emacs* itself. When you need to open another *Emacs* session, instead of loading the whole app, you can run `emacsclient <filename>`. This will attach itself to the current *Emacs* process and open an editing buffer, rather than starting another instance of the app, saving time and memory.

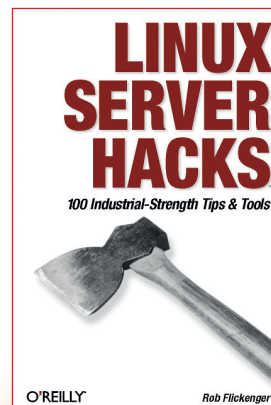


Emacsclient at work. The text in the buffer says it all...

65 Hunt down CPU and RAM hogs

Software: *top* • Difficulty: ■

The *top* command can give you up-to-the-second reporting of system load, memory usage, and CPU utilisation. The simplest way to get started is to



simply run *top* from the command line with the `top` command. You'll be presented with a screenful of information updated every two seconds.

Hit the `?` key while *top* is running to get a list of available commands. *Top* has some very useful display keys – a few of these are `M` (which sorts on resident memory size), `P` (which sorts by CPU usage again), `S` (to toggle cumulative runtime – that is, how long each process and all of its children have been running, in CPU seconds), and `I` (to stop displaying idle processes).

“YOU CAN COMPRESS THE CONNECTION, IMPROVING RESPONSE GREATLY.”

66 Edit remote files locally

Software: *Vim* • Difficulty: ■

Say you have to log in to a remote system and edit a file. Over a slow line, the editing job could be awkward and juddery, but *Vim* offers a quick solution. By passing an `scp://` URI to *Vim*, it'll transfer the file across to the local machine, edit it locally, and then send it back when you're done. A great speedup over dialup lines. An example of editing a remote file is

```
# vim scp://user@hostname.org/home/user/filename
```

67 Copy install CDs to the hard drive

Software: N/A • Difficulty: ■

Many distros need constant access to the installation discs to be able to install new software, which is slow and unnecessary. However, copying all the files off the disc can be an arduous experience. The answer is to make an ISO image out of each disc that you can store then mount on your hard disk. To create an ISO, put the disc in the drive and type this (example for Mandriva) at a shell prompt: `dd if=/dev/cdrom of=MandrivaCD1.iso`.

Once you've created an image of each disc, you just need to mount each image. This should be added to a script so it's performed automatically at boot time. You first need to create a destination directory, followed by the mount command as below; then go to your package manager and configure the mounted ISOs as file sources.

```
# mkdir /mnt/mandriva1
```

```
# mount -o loop -t iso9660 MandrivaCD1.iso /mnt/mandriva1
```

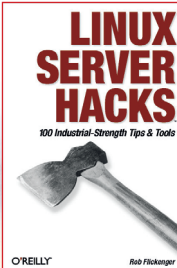
68 Speed up CD duplicating

Software: *cdrecord* • Difficulty: ■■■

The safest method for making a copy of a CD is to first make an ISO and then burn it on to a blank CD. Unfortunately, sometimes you don't have the space (or time) for the interim step of making a copy. If you have a fast enough machine, you can usually burn straight from one CD drive to another, like so:

```
# dd if=/dev/hdb | cdrecord -v speed=12 dev=0,0,0
fs=8m -data -
```

The `-` argument to *cdrecord* means that the data track should be read from STDIN. The *dd* line feeds the *cdrecord* pipe with a copy of the CD in the slave drive on the primary IDE chain (*hdb*). The *fs=8m* parameter makes the write FIFO a bit bigger, to help offset any momentary pipeline hiccups.



71 Boost your filesystems

Software: *mount* • Difficulty: ■■

SUPERFAST TIP



GRAHAM SAYS...

"This tip works particularly well on slower media like compact flash, mounted network drives and large disks on older machines."

By default, every time a file is accessed under Linux an extra piece of data is written to the disk. This indicates when the file was last accessed, which can be useful in some circumstances. Equally, it also means that every file read operation *also* needs a write operation, which is slower, so it has a slight performance impact. You can stop this happening by adding *noatime* to the *mount* options in */etc/fstab*. Beware, though, that a few programs rely on file access times being updated, so look out for problems. Example:

```
/dev/hda1 | ext3 defaults,noatime 0 1
```

69 Get to grips with bash

Software: *bash* shell • Difficulty: ■

Almost every distro uses *bash* as its command-line shell, yet many users don't know of the time-saving features it provides. One of the best is tab completion: instead of typing out every filename and command in full, you can type the first few characters and then press the Tab key for *bash* to complete it. Another handy feature is command history: at the prompt, press the up and down cursor keys to cycle through commands you've entered previously.

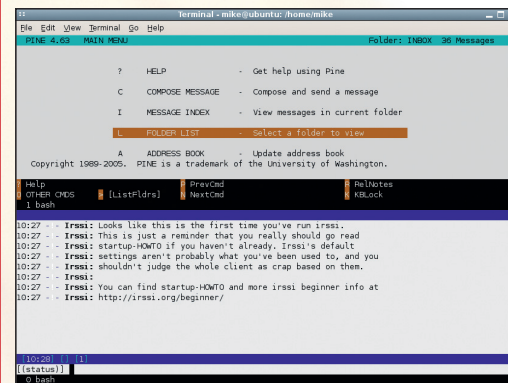
72 Multitask with screen

Software: *screen* • Difficulty: ■



ON the DISC

Screen is effectively a window manager for the text console, allowing you to switch between programs without needing separate logins (on multiple virtual terminals, for instance). You can run, say, *Emacs* on one screen, hit a key and toggle over to *Irssi*. And it gets better: you can detach from the screen process. Find out more about this great tool in our Answers pages this issue (page 102). If it's not in your distro, get it from our coverdisc or at www.gnu.org/software/screen.

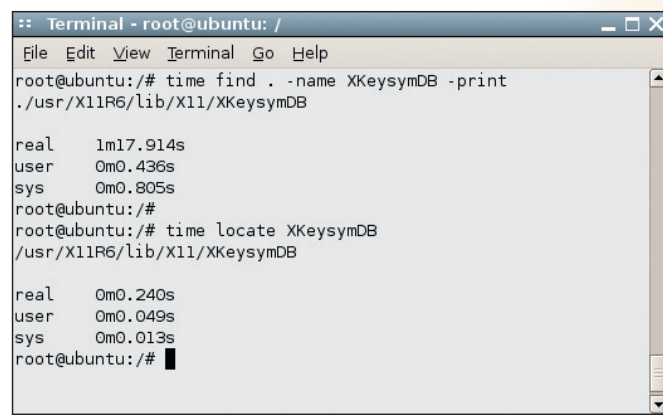


A screen session split into two: *Pine* at the top and *Irssi* beneath.

70 Drop find for slocate

Software: *find*, *slocate* • Difficulty: ■■

The *find* CLI command is useful for locating files but it searches through every single directory rather laboriously. A better solution is *slocate*, which builds a database of file locations for massively faster searching. (The downside is that you need to keep the database up to date). To find out more, open a terminal window and enter *man slocate* and *man updatedb* for the full docs. Many distros configure *slocate*'s database to be updated with *cron* every week.



Time trials: traditional *find* takes over a minute to search through the filesystem, whereas *slocate* just asks its database.

Phew! So there we have it – 72 tips covering all aspects of Linux, and by combining a handful of your favourites you'll see a great performance boost on your machine. Make use of 20 or 30 tips and your PC will be tearing along. If (somehow) you can use all 72 tips on a single box... Well, don't blame

us when you blow the windows out upon reaching Mach 1. But that's not the end: if you've got tips of your own, why not let others know via our forums at www.linuxformat.co.uk? Post your hints and suggestions, and we'll include the best in our next *LXF* online newsletter! **LXF**

OUR THANKS TO...

O'Reilly kindly gave us permission to use some tips from their Hacks books: *Linux Desktop Hacks* (Petreley and Bacon, 2005) and *Linux Server Hacks* (Flickenger, 2003). See www.oreilly.com for more information.