http://www.techworld.com

TECHWORLD

The infrastructure & network knowledge centre

REPORT

Windows vs. Linux

IDG

TECHWORLD REPORT

WINDOWS VS LINUX by David Cartwright

September 2003

Contents

David Cartwright

David Cartwright currently earns a crust as a Norfolk-based technology consultant, specialising in networking, telecomms, security and Internet application development. He has variously worked as a systems and network manager in academia and the defence industry, full-time writer, technical director of an international test lab, head of IT for a multi-national publisher and CTO of an Internet startup and later a venture capital organisation. Since 1996 he has been a member of the judging panel for the UK Networking Industry Awards.

Executive Summary

In the past few years, Linux has moved from being a Geek's plaything to becoming a serious contender as a business operating system. The decision by Munich city council to opt for Linux made the headlines, but this was merely the most high profile addition to the list of Linux successes.

Microsoft has not been idle however. Faced by a serious challenge for the first time for some years, it has responded with gusto and has been determinedly fighting its corner, producing reams of statistics to justify its premier position in the market.

The truth is, of course, that both operating systems have their pros and cons. What has confused the situation is that both technologies have their adherents, some of whom (on the Linux side particularly) can be exceptionally vocal and, it should be said, excessively partisan.

This report tries to separate the myth from reality. The author has examined all aspects of running a modern business system: from the choice of processor to the choice of business application. No aspect of IT is ignored. Subjects covered include web servers, file and print services, networking and access technologies, application software, storage and security.

We offer no blanket conclusion that Linux is better than Windows, or vice versa, but point out the technological areas where one scores highly and the other less well. The IT manager will be able to use this report to examine his or her own setup and assess which operating system would be of most benefit and why.

Windows vs. Linux

Introduction

The Windows vs. Linux debate has run since the late 1990s and shows no sign of abating, just yet. Many contributors to the debate make positive arguments for going down a particular route and it is precisely this reasoned, logical outlook that we look at in this article. Sadly, the discussions are tarnished by the blinkered fanatics who advocate a particular technology for spurious reasons. This is especially so on the Linux side, where individuals have an annoying tendency to think one of Linux's strong selling points as being that "it's not Microsoft".

A thread from a mailing list (full article in the archive at **http://lists.blackcatnetworks.co.uk/main/2002-October/013540.html)** on the subject of Microsoft doing a show at a university, is a typical example of such nonsense:

"I'd be very impressed if anything MS could put on would sway someone who's against MS into liking them."

In this feature, we aim to look at both Linux and the Windows family from a neutral standpoint. Our hope is that a balanced viewpoint will help you make a decision based on the facts, rather than on ill-thought-out opinion. TechWorld is a technology-independent publication and the report's author runs both Windows and Linux on his various servers, with no particular preference.

1. The desktop vs. the server

Linux is making significant progress towards becoming a reasonable choice for corporate desktops. But the lack of commercial application support is a barrier for all but the most basic installations. The greatest boom to Linux's suitability for desktop users came with the release of OpenOffice, an Open Source version of Sun's StarOffice product. This is a complete office suite, with word processor, spreadsheet and presentation package, that goes a long way to emulating Microsoft Office's functionality. Although it doesn't go to the extremes of Office, with things like VBA scripting capabilities, it certainly includes all of the features that 99 percent of office suite users require on a day-to-day basis but at zero cost to the end user.

Sadly though, the enthusiasm of the few is offset by the apathy of the many. Take Macromedia, for instance, whose Flash technology is used by thousands of website owners but is supported in browsers via a plug-in that must be obtained from Macromedia. To its credit, Macromedia lists Linux among its supported clients and there is a downloadable Flash player for Linux. However, Red Hat Linux 9 was announced to the world on 31 March 2003, yet as of the time when this report was written - September 2003 - there is no support in the Flash player installer for Red Hat 9. In fact, you can't install the Flash player without doing a nasty manual bodge. This is because of a bug in Macromedia's installer that won't install the product because it can't find a directory that it doesn't use, anyway.

This isn't a deliberate attempt to single out Macromedia. It's not alone. A while back we were delighted to see a Linux driver CD in the box with an Intel RAID card. We were less impressed that we had to fight for the whole afternoon to persuade the driver to work with our Debian installation because the driver and its associated installer/config programs were Red Hat-specific. Similarly, there has recently been some consternation on the **uk.comp.os.linux** Usenet newsgroup regarding Linux drivers for NetGear wireless cards. Although new versions of Linux, such as Red Hat 9, support the MA401 PCMCIA card, and NetGear has lodged drivers on its website, the noises from the community are that these aren't the latest version. Even with the latest version, you seem to have to make sure you have the latest firmware, which is available only as part of the Windows driver download package.

As long as vendors continue to pay lip service to Linux, but don't go the whole way to providing proper support, Linux won't become a mainstream desktop operating system.

Emulation

You could, of course, use a Windows emulator to run Windows applications on top of Linux. There are several Windows emulators/replacements that will run under Linux (perhaps the most popular being Wine - **http://www.winehq.com/**) often without the need for a copy of Windows to run the actual code (otherwise it'd be a bit pointless). Such systems are generally clunky and don't provide a complete Windows implementation. Because you're running an emulator, performance will also be slower than under native Windows.

Usability

Software support aside, there is a concerted effort in the Linux world to make the look and feel familiar for the user. Because Windows is the leader, and Linux is running to catch up, the makers of "window managers" (the software that puts a GUI on top of the Linux kernel) have gone to great lengths to make the way things work as similar as possible to Windows. Perhaps the most extreme is "qvwm", whose stated aim

(http://www.qvwm.org) is "[to provide] a strong solution for a Windows 95/98-like environment" on Linux and Unix-style systems. Even the window managers that don't set out specifically to look like Microsoft's offerings have many of the same properties; a "Start menu" equivalent in the bottom left corner; similar behaviour when you right-click on items; a taskbar along the bottom of the screen, and so on. There are shedloads of window managers, such as FVWM, MWM, AfterStep, IceWM, Enlightenment etc. The list goes on, seemingly forever. But they're all variations on a theme and that theme feels reassuringly like Windows.

Naturally, there are differences in the applications themselves (although Word under Windows and OpenOffice under Linux have similar functionality, the menu items are different and the shortcut keys differ) but the same issues exist between two similar applications on a single platform.

Remote installation and boot

So we've said that Linux and Windows desktop machines are reasonably similar from a user's point of view, but what about for the tech support guy? Windows has some neat features. For example, Remote Installation Services allow the network manager to create an "image" of a workstation with all the company's standard desktop software and to roll it out simply by network-booting the machine and telling it to acquire its disk image from the server.

Most Linux distributions are network-installable too; one can either boot from a floppy/CD and kick off the installation by hand or, if you're feeling adventurous, you can write an "unattended" installer script instead. Many distributions make this a relatively straightforward process thanks to the inclusion of headless-installer tools, such as Red Hat's Anaconda.

The other main bugbear of network managers is keeping the various software patches that appear regularly up-to-date across a network of desktop machines. Both Windows (via Windows Update and its client-based extensions) and Linux (Red Hat's up2date, Debian's apt-get, etc) have the ability to automatically download and install (either unattended or by asking the user first) updates from the vendors' websites. More importantly, the makers of both systems have realised that someone with 100 identical machines wants to make a single connection to the vendor site and to perform 100 installations over the local LAN, instead of having thousands of potentially lengthy downloads going on each month. For Linux users, what you can do depends on your distribution (Red Hat, for instance, will sell you the Red Hat Network Satellite service for collating updates). Under Windows you can download all the patches you wish and roll them out using computer-based "policies" via the Windows domain service.

System similarities

Linux and Windows aren't all that dissimilar on the desktop, then. From the user's point of view, although the user interfaces aren't identical they make all the same noises. The fundamentals of operating a window-based GUI (clicking, double-clicking, right-clicking, using shortcut keys for menu items, dragging, dropping, and so on) are the same on both systems, so that whichever you're using it won't be a great hardship to understand the basics. Support from commercial application vendors is lagging on the Linux platform but is starting to catch up as more and more of them begin to get it right (something we'll look at later when we consider vendors hardware support in more detail).

Although the bias is more toward Windows on the desktop, the playing field is much more level in the comms room, at the server end of the connection. For this reason we'll be considering Windows and Linux largely in the context of server operations for the rest of this report.

Techworld's conclusion

Linux and Windows are both, from a usability and management point of view, suited to desktop operations. Which one you choose is largely a question of whether you can find applications for your given platform to run the desktop programs (email, word processing, spreadsheets) that you want to use. For a basic office desktop, Linux will save you money; if you can't get the applications you want, though, don't bother with Windows emulation, just buy Windows.

2. Technology facilities

File and print services

We will look into the mechanics of cross-platform file and print services later but Windows and Linux are very well matched in this respect. If all you're looking for is a basic file and print server, with (say) mainstream laser printers and basic Windows file sharing, Linux will do an excellent job for no cost. If you want to start using networked directory services and corporate email address books then it's sorely tempting to decide to use Windows to host these facilities. Given that you get file and print serving thrown in as part of the deal, you'd be silly not to use them under Windows.

Techworld's conclusion

Windows file sharing is native to Windows and in high-end applications will always beat Linux for performance. Where you're not hammering the server, Linux is just as easy and a lot cheaper. Linux's print serving capabilities are pretty well on a par with Windows, except that you have to know more to make it work on Linux.

Server applications

Server applications on Windows and Linux are very closely matched indeed. One of the main functions for which Linux made its name is corporate email. This is not least because commercial Unix-style systems have shipped with a standard email server, Sendmail, pretty much from the year dot, while Windows users have seen competition between Notes, GroupWise and Exchange in the corporate email/groupware arena. Over the years, Exchange has become the clear leader for non-Linux servers. In fact, the Linux world has gone the other way with Sendmail (traditionally regarded as difficult to configure and plagued by security concerns over the years) giving market share to a plethora of new offerings such as Exim **(http://www.exim.org)**, Postfix **(http://www.postfix.org)** and ZMailer **(http://www.zmailer.org)**.

Let's take a moment to look a this diversification of Unix/Linux mailers. Given that Sendmail has always (a) been free of charge and (b) shipped as standard with commercial Unix variants such as Sun's Solaris, it is perhaps surprising that so much competition has arisen. The main reason is that in its early years, until the mid-1990s, Sendmail gained a reputation for containing security holes and for being difficult to configure. It also lacked obvious features such as the ability to handle delivery to multiple domains on the same installation. These other implementations arose simply through the needs of system managers who wanted email server packages that did what they wanted and weren't a nightmare to configure. These days all offerings have pretty much the same features, though in some cases one is slightly ahead of the others (Exim, for instance, makes SMTP authentication protocols rather easier to use than Postfix).

Although Exchange is a commercial (i.e. costly) email package, there are a number of cheaper alternatives for the Windows platform, such as the Merak Mail Server **(http://www.icewarp.com/Products/Merak_Email_Server_Software/)** and the VisNetic Mail Server **(http://www.deerfield.com/products/visnetic_mailserver/)**. Because Microsoft has included support for both native Exchange servers and standards-based POP/SMTP servers, there are no problems with Windows desktop users accessing either Exchange or a third-party server, though with the latter you'll lose the address book function unless the workstations are members of a Windows domain that can preserve a shared user list.

Techworld's conclusion

Linux is a fast email platform, whichever application you use, and Exchange is an overgrown behemoth that's a pain to resurrect if its data gets corrupted. Unless you really want Exchange's advanced stuff (integration with voicemail and fax services, mainly) Linux is the way to go.

Webservers

The other key area in which Linux's popularity boomed is as a Web server platform. In the early 1990s when the Web winked into existence, the organisations making webserver software were writing code predominantly on Unix-style systems, particularly SunOS/Solaris. So, early webservers were almost all Unix-based. What started life as the National Center for Supercomputing Applications' **(http://www.ncsa.uiuc.edu/)** HTTP server evolved into what we currently know as the Apache Web server package. This is, by far, the most popular webserver (Netcraft's Webserver Survey gives Apache 63.72 percent of the market, with Microsoft Internet Information Services at 25.95 percent). Until version 2.0 this didn't even formally support Windows except in pre-release form. With Windows NT 4, Microsoft introduced IIS, which is now at version 6.0 with the incarnation that ships with Windows Server 2003.

The platform to choose is largely dictated by what languages you want to use. IIS is inherently linked with Microsoft's ASP (a VBScript/Jscript-based programming language) and ASP.NET (the new evolution of ASP that follows the new .NET language structure and uses the .NET Common Language Runtime). Linux-based Apache servers, on the other hand, are more suited to traditional CGI scripting (the original way to make a Web server programmable, with scripts traditionally written in Perl or C/C++) or the flourishing PHP (a derivative of Perl that makes it far easier to integrate into Web pages than its predecessor).

Techword's conclusion

If you want to use ASP and ASP.NET, go with Windows. Although you can run ASP on Linux (http://www.chilisoft.com) you really don't want to, as you can't access the useful add-ons that third party vendors make, which are universally Windows-only. Although there are moves to make an Open Source .NET Framework equivalent, it's a long way off usable right now. If, on the other hand, you don't care what language you use, Linux gives plenty of choice. The fact that Apache has roughly three times the installed base of Microsoft IIS must count for something.

Databases

The programming language of choice is more important than, say, the back-end database choice. Because most Web systems are now client-server you can have the pages running on one platform and the database on another. It goes without saying that Microsoft SQL Server is an almost automatic choice for Windows-based websites simply because the SQL Server connections between ASP and SQL Server are native Microsoft calls that don't go through some intermediate layer, such as ODBC. That said, though, it's perfectly feasible to integrate (say) a Windows-based webserver running ASP with a Linux database server running MySQL **(http://www.mysql.com)** or PostgreSQL **(http://www.postgresql.org)**, as there are Windows ODBC client drivers for both of these databases (and many others). Oracle **(http://www.oracle.com)** is also something of a guiding star for cross-platform Web systems, as it produces native versions of its database system for both Windows and Linux clients and servers, allowing native (i.e. fast) database connectivity between client and server instead of making the programmer resort to ODBC with its relative sluggishness.

Leaving aside Web-backend databases for a moment, commercial DBMSs are among the most advanced in terms of cross-platform support. We've already mentioned Oracle's abilities to work on Windows and Linux, but other database manufacturers such as Sybase and IBM (with its DB2 product) are supporting Linux to some extent. In Sybase's case, for instance, the basic database products are available for Linux but some advanced options aren't.

Techworld's conclusion

SQL Server is Windows-only and is an attractive database server, but other commercial DBMSs such as Oracle run on both Windows and Linux. Although the list prices are higher than SQL Server, discounts bring them back into line. There are also plenty of free databases on Linux (MySQL, PostgreSQL), though one shouldn't forget MSDE on Windows (a cut-down version of SQL Server that ships with MS Office Professional and is free to use) as the budget option.

Real-time streaming

Video and audio streaming, although hardware- and bandwidth-intensive, are growing in popularity. Microsoft has its own Windows Media streaming server, which simply drops onto a Windows server. The other main player in the game, Real Networks, offers server support for Windows, a variety of commercial Unixes, and Linux. Real's product specs flag an interesting point. Instead of, as is often the case, stating that a software product is compatible with a particular version of a particular Linux distribution (e.g. Debian 3.0), Real Networks' Helix Universal Server doesn't mention any specific distributions. It simply states that you need at least version 2.4.18 of the Linux kernel and release 2.2.4, or better. These are low-level software options upon which the software relies. It can sometimes take a bit of research on the Linux distributors' websites to find out the revision levels of the underlying components.

Techworld's conclusion

For streaming, there's a great deal of balance between the two platforms. So, unless your OS decision is made for you by a strange desire to use Windows Media Server, it's really a matter of personal preference.

Backup software

So far we've encouraged a cross-platform implementation. It's perfectly feasible, for instance, to use Windows-based webservers talking to Linux-based databases, or have your users running Microsoft Outlook on the desktop but use a Linux-based mail server to keep the costs down. There is a key issue with such a heterogeneous setup, though. At some point you'll want to back everything up.

Generally speaking, Windows systems (particularly from Windows 2000 onwards) are more complex to back up than Linux systems, because the "system state" information (registry entries, COM/COM+ entity registrations, and such like) aren't represented simply as files on the disk (whereas on Linux, everything's just a file or a directory).

The most expensive choice is to have two backup systems, one for Linux and one for Windows. Although the software costs will probably be cheaper if you use Linux (the de facto standard is Amanda **http://www.amanda.org** which is an excellent package and costs nothing) you'll need to have a tape drive in both worlds. Although you could conceivably back up your Linux filestores to a holding disk on a server, and then have the Windows backup world copy these files over the network to its own tape drive, this is clunky to say the least.

Most mainstream Windows backup systems have similar features, notably the ability to back up several servers to a single tape drive; to back up open files; and to handle real-time applications, such as Microsoft Exchange and SQL Server, without the need to close the service down while the backup is executing. (The built-in backup application that comes with Windows can handle file/folder backups, can save the system state information, and has an in-built scheduler for unattended backups, but doesn't do the advanced stuff such as open file copying). Generally, Linux filestores are not supported by the entry-level systems, though companies such as Legato, with its more high-end NetWorker backup package, do include Linux. Although there are both server and client options, they are only available for a handful of distributions (specifically Red Hat and SuSE with NetWorker).

The freeware Linux backup packages don't have support for open files, specific databases, and so on, so these can be an issue to deal with. On both Windows and Linux there are alternatives to using application-specific backup software add-ons; namely to stop the application(s) in question for a short time, make a quick "snapshot" of their files, start the application up again, then back up the snapshot. Most mainstream applications on both platforms have built-in backup options. You can schedule these via either the application's own scheduler (e.g. SQL Server 2000) or the system scheduler (e.g. anything on Linux).

The backup software issue is perhaps the server world's greatest similarity with the desktop world as commercial software vendors haven't yet grasped the nettle and said: "We'll support Linux just like we support Windows".

Techworld's conclusion

The mainstream backup packages are better on Windows than on Linux, as they have downtime-free capabilities for open files and 24x7 applications. The higher-end commercial ones work on both platforms. The free ones for Linux can't do open file backups, but in many setups it's no great problem to run a scheduled script to do a quick export of the data to a file that the backup software can then read at its leisure.

Disk subsystem support

Windows' support for modern disk subsystems is, it's fair to say, universal. Except with mainframe manufacturers, when a new type of storage system is developed it is automatically made to work with Windows, because that's where the marketplace exists. Although perfectly happy with well-established concepts such as SCSI and ATA (IDE) disks, Linux is less up-to-date with its support for the more modern filesystems. Storage system support is a key part of the Linux kernel and as such it is the kernel development community that produces the appropriate drivers. Because SCSI is such a long-established technology there are well over 50 chipset-specific drivers shipped with the Linux kernel today (which equates to many hundreds of adaptors in the real world, as there are often many different adaptors using the same chipset). Likewise the popular RAID cards, such as the Compaq SMART2 and the Mylex DAC960, are supported out of the box.

Although you often find that Linux supports the better-known kit (the Compaq and Mylex RAID example is typical) this doesn't mean that if you want to use a spanking new technology (or just something that's a bit off the wall) you're definitely out of luck. Serial-ATA isn't exactly mainstream technology but there are Linux drivers for some adaptors, such as the 3Ware Escalade, for example. If you're an iSCSI fiend then you'll find **http://linux-iscsi.sourceforge.net/**, a project dedicated to implementing the iSCSI standard on Linux, of interest.

This latter example raises another good point. When something becomes a standard, the Open Source community tends to jump on it and make a decent job of implementing it, for free, for the masses. This is understandable, because there's a much smaller danger of your efforts being in vain when you're implementing a standard, than when you're trying to make a driver for something proprietary that may not catch the industry's imagination. This said, though, the Linux developers should be applauded for the fact that they do get to work quickly on their implementations, once it looks like a proprietary technology is starting to be used reasonably widely.

Moving back to RAID, for a moment, both Windows and Linux have software RAID options (i.e. the operating system not hardware does the mirroring/striping/partity tasks) that are relatively simple to use. For a long time it was incredibly difficult to persuade a Linux machine to boot from a partition that was part of a software RAID cluster but as technology has progressed it's less of an issue.

Techworld's conclusion

Anecdotal evidence has it that Linux's software RAID implementation is second to none and so it's a fine choice if you can't afford a RAID option. If you want the hardware option, then pick a popular card and you'll have a choice of operating systems.

Network technologies

All network adaptors ship with Windows drivers. The majority of them even ship with DOS drivers if you're that way inclined. It's rare to see a vendor supplying a Linux driver with an adaptor (though if, for example, you're a NetGear customer running an MA401 wireless LAN card you're in a happy minority). As with the SCSI and RAID section, above, though, if you stick with a mainstream name there won't be a problem under Linux. There is an issue, though, with some on-board Ethernet controllers where the network hardware is part of the gubbins that's soldered to the motherboard. You're stuck with the chipset that the motherboard manufacturer decided to buy, which could well be a weird and wonderful Taiwanese make nobody (particularly a Linux driver writer) has ever heard of.

Although we mentioned NetGear's WLAN drivers above, all isn't sweetness and light on the IEEE802.11 front. Many mainstream WLAN adaptors are either listed as experimental drivers or not supported at all. The usual Linux rules apply: the newer it is, the less likely there is to be a driver for it.

Access technologies

Wide area networking is reasonably well supported in both Windows and Linux. As usual, any adaptor you buy will have a Windows driver on the CD in the box, but Linux does OK here. For popular proprietary adaptors, such as the Alcatel SpeedTouch ADSL modem, drivers have been around for some time and work just fine (though installation can be fiddly). For traditional modems life is peachy, because pretty well any modem that supports the de facto standard Hayes "AT" command set will work with Linux's PPP (dial-up) drivers. Back in days of old, US Robotics invented something called a "WinModem", which was an internal modem card that relied on the PC hardware and Windows software drivers to provide some of the device's functionality instead of building this functionality into the modem itself. Such devices have never made it into the Linux world because the manufacturers often don't produce them and refuse to license their designs or code to third parties who would be only too happy to do the work. WinModems are still common, as they're cheap to manufacture and buy, but there are plenty of "full feature" modems on the market for as little as £25-£30, that will work fine with both Windows and Linux.

ISDN is slightly more complicated than modem connectivity, but because it's such an old technology, it's well supported by today's Linux kernel. The bonus with ISDN is that there's a lot of similarity between it and modem operation, though there are some extra features to cater for, such as channel bonding.

Techworld's conclusion

There's a excellent chance of all but the weirdest network adaptors running with Linux. If you have a WinModem, it won't work with Linux but it'll only cost you £30 for a proper one that will.

Tape subsystems

We mentioned backup issues earlier, but one of the main concerns with performing backups is actually connecting the backup drive (generally a tape device) to the server and making the two communicate. At the basic level this is really down to the SCSI or Fibre Channel driver, because controlling the device itself is mainly down to the backup software, not the capabilities of the operating system. Although SCSI is well catered for by both operating systems, Fibre Channel is pretty sketchily supported in Linux, with the Interphase 5526 **(http://www.iphase.com/products/specificProduct.cfm/1/1/16)** being the only real option.

Support for tape autoloaders, and other more advanced gadgets, is down to the backup application itself. Even freeware packages, such as Amanda, are able to cater for the majority of popular tape changers, though if you're willing to spend money on an autoloader the cost of the backup package itself shouldn't be too much of a concern.

Techworld's conclusion

The tape drive hardware itself isn't a problem even if Amanda doesn't list it specifically. You can generally build your own configuration anyway by copying a similar one and modifying it. Tape device support is really down to the SCSI card it's connecting through.

Offline storage

While Windows provides support for removable media and their management in an HSM (Hierarchical Storage Management) setup, Linux is lacking in this respect. Although the kernel includes an excellent volume manager for online (permanent) disks, support for HSM is extremely limited even in the commercial arena. So if you're looking to use removable, offline media to any extent, Windows is probably the better road to go down.

Techworld's conclusion

Windows is really the only sensible choice if you're into offline storage in a big way.

Networked printers

One of the problems with modern printers is that they're just so darned complicated. With driver software packages stretching to many megabytes, particularly for multi-function printers that can also play at being fax machines or scanners, it's no surprise that there aren't any Linux drivers for a lot of the more esoteric devices. That said, there are plenty of common printer control languages (PCLs) out there, not least PostScript (an Adobe creation), PCL-5 (HP's proprietary PCL) and the Epson ESC-P protocol. So, if the printer you're trying to connect to supports any, or all, of the above (and even low-end printers tend to support one of the basic standards) then you stand a good chance with Linux. In fact, if you have a heterogeneous network Linux may well be a better option for running your print queues as many Linux incarnations are able to handle locally-connected (parallel or USB), Unix-style LPD, Windows and NetWare printer connections, as well as talking direct to HP JetDirect devices (Hewlett-Packard's proprietary printer-to-Ethernet connection boxes).

The advent of the Internet Printing Protocol **(http://src.doc.ic.ac.uk/rfc/rfc2910.txt)**, largely through the collaboration of printer manufacturers such as HP and Xerox and supported by both Linux and Windows, has made the communication from client to print server somewhat simpler. Companies such as ESP **(http://www.easysw.com/printpro/)** are addressing the issue of device-specific drivers for Linux (and other Unix-style) clients.

Techworld's conclusion

Printers aren't a problem on either Linux or Windows, though Linux can be more difficult to set up. If you insist on having multi-function printer/scanner/fax devices, Windows is the safest option.

Multi-processor support

Symmetric multi-processing is available with most Linux distributions and in the kernel distribution too (though as most people have single-processor machines it's turned off, by default, in the kernel build menus). Windows also supports SMP, though the version you have dictates the maximum number of processors; you can use four for Windows 2000 Server or Windows Server 2003 Standard Edition; eight for Advanced/Enterprise; and 32 for Datacenter. Windows Server 2003 Web Edition, which has no Windows 2000 equivalent, is restricted to two processors.

Techworld's conclusion

SMP support is excellent on both Windows and Linux. Interestingly, the SMP code of Linux is one of SCO's bugbears (a small portion of the Linux SMP software is, the company alleges, infringing its Unix copyright).

Resource requirements and restrictions

With Windows, the version you have dictates how much resource you can put in the machine. Windows Server 2003 Web Edition has a limit of 2GB RAM, while the Standard, Advanced and Datacenter editions can handle 4GB, 32GB and 64GB respectively. (Note that if you're using the versions of Windows for IA-64 processors, the Enterprise Edition can take up to 64GB RAM and the Datacenter Edition up to 512GB). The theoretical disk partition size limit with Windows is 16 Exobytes (264 bytes) so long as you're using NTFS partitions. Linux, with a 16GB RAM limitation (again, there are add-ons you can use to get this figure higher) can handle up to a Terabyte in a single disk partition without problems (and there are kernel patches that let you go further if you're so inclined). As far as CPUs are concerned, you can go up to 16 with the standard kernel. In practice, the majority of installations stop at around four processors, because the overhead of inter-processor communication becomes prohibitive and the speed gained from adding more processors becomes less and less significant.

Techworld's conclusion

For the average installation, resource requirements don't get in the way. Unless you're getting really ambitious, you don't even need to ask the question. Windows' limits are higher, but you're getting into real top-end stuff there (for instance, you probably wouldn't be buying WS2003 Enterprise Edition because you wanted more RAM, you'd be buying it because you wanted clustering and Standard Edition can't do it).

IPv4 and IPv6

Traditional IP connectivity is supported out of the box by both Windows and Linux. IPv6, the new release of the Internet Protocol designed to get around the problem of the world running out of addresses, is also supported for the first time by Microsoft in Windows Server 2003 (and Windows XP, for desktop users). More information is available at **http://research.microsoft.com/msripv6/.** Linux has experimental support for IPv6 in the kernel, but has yet to extend into the standard, full release product. That said, many of the applications that underpin the Internet, such as the BIND domain name server that ships with most Linuxes, are ready to work with IPv6, but are just waiting for the world to need it.

Techworld's conclusion

Windows is ahead of Linux on IPv6 support, but quite frankly, who really cares?

Non-IP network protocols

For those who think there's more to life than TCP/IP, both Windows and Linux provide support for network protocols that used to be popular a few years ago. You can turn on IP, DECnet, AppleTalk, 802.2 LLC, X.25, even good old Acorn Econet (maybe someone's got an old Archimedes or BBC Micro out there). Microsoft has decided to chuck away many of the lesser-used ones (such as DECnet), but still support AppleTalk, IPX and NetBEUI (the original transport for NetBIOS traffic).

Techworld's conclusion

Linux is nice if you're into antique network protocols.

Remote management

When we covered desktop issues earlier, we mentioned that both Windows and Linux have a number of remote management and installation capabilities. Windows is perhaps the better evolved of the two systems in this latter respect, simply because of the tight coupling provided by its "domain" system, in which desktops and servers are securely connected and policies can be defined per-user or per-machine. This means that whenever a workstation is turned on or rebooted, the domain server can pass out software updates or configuration changes, and whenever a user logs in the options, directories, files and applications he or she has access to can be controlled in minute detail by the system administrator. Windows' concept of "roaming profiles" also allows a user's personal settings (and even the contents of their "My Documents" folder) to follow them around to whatever PC they happen to be sitting at. In reality, the latter concept can be a pain (if you keep gigabytes of junk on your desktop, be prepared for a huge file copy when you log into a machine you've not used before) but can be extremely useful if used wisely.

Linux doesn't allow the network manager to lock everything down in the way that Windows does. It does, of course, have the same benefits as Windows servers of controlling access on a per-user basis. With add-on software a Linux workstation can become part of a Windows domain authentication system, a Novell NDS tree, or a native Unix-style networked authentication scheme such as NIS (a Sun invention, previously called Yellow Pages until BT's trademark people got all litigious). Also, because all Unix-style systems are inherently multi-user, the system administrator can connect from afar and perform changes and installations without interrupting the user's session.

Although Linux doesn't allow you to lock stuff down in the way Windows does, this doesn't mean you can't provide any level of lock-down. When Unix was first introduced, hardware was inordinately expensive (tens of thousands of pounds for even a basic Sun workstation in the early days) and so it made sense to have one set of disks in the server room and to have diskless workstations booting from a server and using the network as their path to their hard disks. This capability hasn't died out in Linux, and so you get what is perhaps the ultimate in manageability you have a single software image at the centre of the network and everything boots from it.

With regard to remote administration of computers, though, Linux is streets ahead of Windows simply because of its multi-user heritage. Put simply, there's nothing you can do sitting in front of a Linux computer that you can't do over a network from hundreds of miles away, because you simply log into the box and go for it. Admittedly Microsoft has addressed this issue to an extent, with the introduction of Terminal Services-like functionality in Windows XP Professional and all of its Windows Server products that allow remote control. However, this relies on a reasonably fast link existing between the two endpoints as it's a GUI-based system - thus a lot of traffic flows to and fro with screen refresh data. While you can do the GUI thing with Linux, you don't have to. Everything on a Linux machine was designed be managed on the command line. Linux's other bonus in this respect is that since most application configuration files are simply chunks of text, it's easy to copy a machine's configuration back to the administrator's PC so he can try it out in a controlled environment. In addition, the administrator can make a copy of the old settings and then fiddle to his heart's content in the knowledge that if everything goes wrong, he simply needs to copy the old settings back and restart the application.

In the late 1990s, the likes of IBM, Sun and NCD decided that "thin clients", which used this remote boot facility to download their operating system images over the LAN, would be a neat idea. Although thin clients are in the minority of new PCs these days, systems such as Citrix MetaFrame **(http://www.citrix.com/site/PS/index.asp)** and Microsoft Terminal Services allow this remote-control type of operation in a Windows setup.

Techworld's conclusion

Windows scores well on networked software distribution with its system of policies, and Terminal Services is an excellent bit of kit. Since Linux and its Unix peers has always been inherently networked and multi-user, it romps home on its remote management features.

Security

Security under both Windows and Linux is dealt with via a set of users and groups, each of which has defined permissions to see, open and change each file. Both systems enable both local (single-machine) or networked (multi-machine) login files to be used, and via standards-based protocols such as LDAP it is possible for any number of different directory services to interconnect. File and directory security under Linux is split into three categories: user-level, group-level and everyone. Within each level, an entity can be given any or all of read, write and execute permission to any file; the "execute" permission on a directory dictates whether a user can list the files in that directory or not. There are also extra permissions that allow you to dictate that (say) a particular program should execute under a given user ID instead of as the currently logged-in user. Windows has similar permissions to the standard Unix security measures, with the addition of a "modify" flag alongside the "write" flag, which gives a little more control over file deletions. Under both systems, it's possible to recursively allocate folder permissions to all subfolders.

Leaving aside general user access control for a moment, both systems have application security issues. If you're running a Web server, or FTP server, or any type of networked protocol, you need to watch for security holes. So there is a great deal of application-level protection in both systems. Linux is a tad stronger than Windows with its "chroot" concept, a kernel-level security feature that, when applied to any application, makes that program believe its top-level data directory is the top of the main directory structure. This effectively prevents an intruder doing "cd ..", the technique used by the recent Unicode attacks on Web servers. The fact that Linux is so inherently multi-user is a potential security issue, and so traditional unencrypted remote access methods, such as Telnet, are gradually being replaced with SSL-based (Secure Sockets Layer the encryption/authentication mechanism used by secure webservers) terminal programs.

At the other end of the security spectrum, there are times when you have to run programs as the "administrator" or "super" user.. Backup packages, for instance, have to be able to access all files on the system in order to copy them to tape. Windows allows trusted programs to be installed by the system manager as "services", which run under a stated user ID instead of executing as the current user. Linux has the same concept, except they're called "daemons", and both Linux and Windows allow these packages to be started and stopped at will by the system administrator. Both systems also allow the system admin to be logged in as a normal user and execute programs under a privileged ID. Windows uses the "Run as..." facility, whilst Linux has the "su" command that allows you to switch immediately to the super-user ID.

Techworld's conclusion

Although the two systems handle security in slightly different ways, they're both perfectly adequate for the job. It's not a factor you'd use in the decision of what to buy, because they both tick the "sufficient security control and enforcement" box.

System logging/reporting

Since its inception, the BSD Unix family (and anything else, including Linux, that thought it was a good idea) has had the "syslog" service, which is a networked, highly-configurable logging mechanism whose facilities overshadow those of the Windows event log. The latter is useful, and applications can create their own event log streams outside the standard Application, System and Security trio, but the much more flexible and longer-evolved Unix/Linux syslog concept **(http://src.doc.ic.ac.uk/rfc/rfc3164.txt)** takes it much, much further and makes log data way more useful and usable. First of all, log entries can easily be redirected to a central "log host" for collation. More importantly, it's possible to tell the system to send different types of log message (information, debug, error, and so on) to different files. This means that you can maintain a standard set of logs that potter along day in, day out, and then occasionally turn on debugging on one service or other. You can see the results not in amongst the day-to-day messages but in a separate file, perhaps in a completely different place. The syslog protocol is also supported by many network devices (particularly firewalls) and so you can use your Linux server as a central collation point for log entries from devices all across the network.

Techworld's conclusion

Linux (and Unix) logging eats Windows for breakfast.

An embedded platform

We have already touched on the fact that both Windows and Linux are very useful as the basis for networked applications such as Web serving or email transport. But some applications need the operating system more as an embedded system than a generic interface between the program and the hardware.

Check Point FireWall-1 **(http://www.checkpoint.com)**, for instance, sits on top of Windows 2000 (and some Unix incarnations such as Solaris, incidentally) and provides security functionality for the corporate network. It begins with a native Windows installation and installs its own "hooks" into the various system functions in order to ensure unwanted connections and traffic don't make it into (or out of) the enterprise. Linux, on the other hand, because it is an Open Source system, is often incorporated into the firewall program itself. An obvious example is the SmoothWall Corporate Server **(http://www.smoothwall.co.uk/)** which has an embedded Linux kernel and installs straight onto the hardware instead of needing the OS to be installed first. The latter approach is potentially more secure because the firewall writers can chop out the bits they don't need (and which, if left in, could have bugs that make them a security risk) and because they have full access to the source code of every last part of the OS. With Windows, the developer has to rely on Microsoft to provide (a) the guarantees of security; and (b) correct documentation for all of the complex, low-level hooks the firewall needs to link into.

Even on a level playing field, where an application sits on top of the operating system instead of including the operating system within the application itself, Linux provides much lower-level (and, importantly, much more open, source- and documentation-wise) calls to system networking functions. The kernel has deliberately had such calls built in over the years precislely so that it can even be used as a basic packet-filtering firewall in its own right without any programs sat on top. Up to and including Windows NT 4, installing a firewall package (or anything that needs to do low-level networking stuff, for that matter) has required a convoluted, hacky-looking installation process, although life seems rather more streamlined under Windows 2000, which has clearly been designed with the needs of low-level networking programmers more in mind.

Techworld's conclusion

Linux is sufficiently low-level for it to be immensely useful both as an embedded OS and as a system on top of which applications can sit whilst simultaneously delving into low-level system calls. Windows has traditionally been running to keep up, and it's almost there now.

Network management

Strangely, neither Linux nor Windows is particularly well equipped out of the box for monitoring what is going on in the enterprise. This is largely left to the applications to achieve mainly by logging activity via the provided log functionality although both systems have basic process resource analysis, network sniffing and such like. On the positive side, there is a vast amount of software available for each platform that the network manager can bolt on to make the management task simpler. Packages such as HP OpenView **(http://www.hp.com)** and Heroix RoboMon **(http://www.heroix.com)** are commercial, Windows-based network management offerings; on the Linux side there are dozens of freeware and inexpensive tools for packet analysis (Ethereal being the most obvious, **http://www.ethereal.com/**), network monitoring and remote reporting (such as Big Brother, **http://www.bb4.com**). Although perhaps not as big and integrated as the commercial Windows offerings, they provide significant benefits for not much cash. Note that many of these Linux/Unix-orientated offerings also extend into the Windows world, including the two mentioned here.

Techworld's conclusion

For management applications, Windows wins simply through application support. For the underlying reporting capability, Linux leads but doesn't really have the range of commercial applications to collate all that useful reporting information.

3. Hardware support and drivers

Anyone who has bought a new model of computer, graphics card, tape drive, SCSI card or other piece of computer hardware and has then tried to install Linux on it will generally have found that Linux lags behind Windows in terms of the availability of drivers.

In some cases it's not that there isn't a driver for the product but that the standard installation doesn't support a particular configuration - such as when adding a new monitor with a different refresh rate or screen dimension from its predecessor. Because Linux tends to work with text-based configurations files (though there are more and more GUI-based applications that plonk a friendly front end onto these often intricate masses of unintelligible text) it's often impossible for the average system manager to comprehend how to change the settings he needs to cater for the new equipment. In many cases, in fact, it's not even obvious where to look for the config file, because one of Unix/Linux's failings is that configuration data is scattered all over. Contrast this with the way Windows works. Everything's in a fairly obvious place and it's all point-and-click and user-proofed with "You can't do that refresh rate on that monitor" warnings. It's no surprise that if you've got new hardware, you stand a better chance of it working properly with Windows.

Part of the reason, of course, is that the CD that came with your new piece of kit has drivers for all the various flavours of Windows on it, but almost certainly doesn't include drivers for Linux machines. There are a few exceptions to the rule. Intel, for example, has bitten the bullet and ships Linux drivers with many of its RAID cards, though last time we tried one, we had some fiddling to do because we were using Debian Linux and the driver was designed for Red Hat. In general, though, Linux is a noticeable omission among the NT, Windows 98, NetWare and even DOS drivers on the enclosed CD.

Some vendors are trying, though. We mentioned earlier that you can get a Linux driver for the NetGear MA401 wireless LAN card, and indeed it's sitting there on the company's website right now. (It's also supported as standard by recent Linux kernels). Adaptec is another Linux-friendly company. Its support page lists driver downloads by product or by OS, and there's a page full of assorted Linux drivers. To pick another big name at random, Cisco is also making the right noises. The url **http://www.cisco.com/pcgi-bin/tablebuild.pl/aironet-utils-linux** marks the page for a WLAN adaptor we didn't think was supported by Linux but which has a Linux-specific driver available from Cisco itself.

Other manufacturers have decided to go halfway and instead of making drivers themselves, work with third parties to help them build the drivers for them. HP, for instance, has announced **(http://h20015.www2.hp.com/en/document.jhtml?lc=en&docName=bpu00659&cc=us)** that it's working with developers to enable them to develop Linux drivers for the all-in-one printer/scanner/fax products we hinted at when talking about printer support issues.

Many vendors are, of course, not exactly going out of their way to help with the Linux driver development process. The various "How To" documents we've seen on the Internet concerning getting the Alcatel SpeedTouch ADSL modem to work with Linux, generally bemoan the fact that as well as getting the Open Source part of the driver you still have to go to Alcatel's website, register your name, and download a software component because the company doesn't want to allow third parties to allow it to be freely downloaded from external sites. Although vendors have become more co-operative over the past few years, the Linux driver developer sites used to be rife with comments along the lines of "I can't do a driver for card XX because the vendors won't tell me anything about the architecture or what calls I have to make."

Driver quality

Given that many Linux drivers are written by people out of the kindness of their hearts, without financial incentive, most are surprisingly stable. As vendors become more co-operative with driver writers, driver bugs are reducing. People who previously had to try to reverse-engineer code and find API calls by trial and error are now often able to obtain proper technical information about the hardware and can thus be more confident about having a driver that handles all possible extreme cases, interrupts and so on. This means fewer crashes for the user and more confidence for the network manager (and, more importantly, the board). There is, however, a natural tendency to be nervous of any third-party driver not least when you read on the vendor's download page that driver X will only work with version 23.34.45 of the adaptor's firmware, running on a given version of Windows Server 2003 with a particular patch installed. In general, though, Linux drivers, even though they have often been written without vendor documentation, are stable and reliable.

Techworld's conclusion

Hardware support isn't a reason for choosing an operating system. The lack of support for a particular device that you absolutely have to use, sometimes because it's the only one of its kind but often for political corporate-standard reasons, is a show-stopper, though. So check the hardware compatibility lists, and only discount one or other operating system if it doesn't support the kit you're planning to use (and, of course, only if you're precluded from making a different choice of hardware that would make the compatibility issues less significant).

4. Version control

Linux and Windows are both a nightmare of versions and releases, albeit in different ways. Windows, for instance, is available as Windows 2000, Windows XP or Windows Server 2003, which sounds simple until you realise that Windows 2000 has three editions (Standard, Advanced and Datacenter), Windows XP has two (Home and Professional) and WS2003 has four (Web, Standard, Enterprise and Datacenter). That adds up to eight different flavours, each with its own peculiarities. Internet Information Services, for instance, can run on Windows XP but you can't have multiple named servers, whereas on either of the Server products this setup is possible. Similarly if you're running WS2003 Web Edition then you can't make the server an Active Directory controller, but with the other three incarnations you can.

Linux versions

Linux's confusion, on the other hand, is down to the sheer variety of distributions that exist; SlackWare, Mandrake, Red Hat, SuSE and Debian to name just five. Many distributions also have a number of versions. Mandrake has four, for instance, plus a dedicated firewall option. The differences between the various distributions are explained by the actual nature of a Linux distribution. That is, each distributor takes the latest stable version of the Linux kernel and its associated libraries, bundles together as many Open Source software applications (games, compilers, database server, office automation products, X-Windows window managers) as can reasonably get, and then adds some of its own custom gizmos. The two most obvious differences between the distributions, then, are the installers (Red Hat, Mandrake and Caldera have historically been excellent and largely idiot proof, while Debian's was notoriously cryptic) the GUI-based config tools (which hide the text-based configuration files from the user) and the auto-updaters that pull down the new versions of the kernel and the associated applications (an area where Debian led the field until the others, Red Hat in particular, caught up in the last couple of versions). Because the distributions include as many associated Open Source packages as the distributors could find, the actual range of programs available to the user doesn't vary much from distribution to distribution.

Keeping up with the changes

The vast range of different incarnations of what one initially thought were actually the same thing mean that software vendors have a hard time ensuring that their products continue to be compatible with new releases of the various operating systems. Fortunately, the vendors generally maintain the underlying functionality of the OS unmolested, concentrating mainly on bugfixes, and when a new version of a package is produced, the system calls are kept the same and the program's files stored in the same place on the disk, and so it's usually possible for the application writers to be confident their programs will keep working.

The main issue is when a new version of an OS, or distribution, arrives. Each of the Linux incarnations should be regarded as a new product, because files are likely to have moved between releases. So if you want to install the current release of SunONE ASP on a Red Hat 9 machine, forget it because it was designed for a previous version and although it'll install, it doesn't work. From one release of Red Hat to the next, for instance, the changes to the layout of files on the disk could have changed almost as much as during the leap from Windows 2000 to WS2003 and the chances are, therefore, that any application that relies on certain files or directories existing in a particular place will simply not work.

Of course, the same applies when moving from one release of Windows to another. Although there are minimum requirements specified in terms of service packs (e.g. some applications wouldn't install on Windows

NT 4 unless you had at least Service Pack 3) you won't find that something that works on NT4 SP2 doesn't work on NT4 SP3. Try running a Windows 2000 anti-virus package on WS2003, though, and you stand a good chance of the installer saying: "No way, I don't like this version" Similarly, the software CD shipped with our HP OfficeJet clearly says: "Don't install me on Windows XP, go to the website and download another version." Another example is Norton AntiVirus 2000, one of whose components had to be updated from the vendor's website before the package would work on Windows XP.

Version support

Whether you use Linux or Windows, the package you use will generally receive support from the vendor even if it has been superseded by a new version. Microsoft has a formal timeline **(http://support.microsoft.com/default.aspx?id=fh;en-gb;lifecycle)** of its product support, which tells you when it will stop making new bits for the package and provide bugfixes only, along with when it will stop supporting the product altogether. So Windows 2000 Server will receive updates until the end of March 2005, whilst WS2003 will be supported until June 2008.

The Linux vendors are generally less formal about their product lifecycles, but as with Microsoft there is a significant overlap between support for one version and support for the next (and even the next one or two after that). That is, with either system you can be confident that just because the next release is out, the current release hasn't been thrown in the bin just yet. Do bear in mind, though, that Linux distributors have less cash to throw at support than Microsoft, so the lifetime of a given Linux distribution will be significantly shorter than that of a Windows implementation (Red Hat's typical product lifetime, for example, has been one year, though the company announced earlier this year that this figure is likely to be revised).

Platform support

Back in the Windows NT 4 days, Microsoft supported the DEC (now Compaq) Alpha processor as well as the Intel 80x86 range. Alpha support went by the wayside with the release of Windows 2000, and although there was some consternation from the users, the decision was final. So although Microsoft supports Intel's new 64-bit IA-64 processor with its high-end products (notably WS2003 Enterprise and Datacenter Editions) it is firmly an Intel-only house.

Linux, on the other hand, continues to support a vast number of different architectures. The Linux kernel will compile happily on Alpha, ARM, PA-RISC, IBM S/390, Intel IA-64, Intel 80x86, Motorola 680x0, MIPS, PowerPC and Sun SPARC, and while many vendors (Red Hat included) concentrate on the mainstream Intel 80x86 implementation, Debian in particular stands out by supporting all of the above architectures.

Such support provides an interesting migration path for owners of Alpha kit which is, by the nature of its architecture, still relatively high performance equipment, but which can't do anything more advanced than an aged (and obsolete) version of Windows.

Special editions

We have already mentioned the Datacenter versions of the various Windows server products. These are highly specialised releases designed for very high-end, niche applications, and as such have specialised support teams to deal with the very small number of specialised clients.

The Linux community also has its share of specialist, high-end implementations of Linux, although in reality they are more like Windows, Advanced or Enterprise versions than the massive Datacenter edition. Red Hat Enterprise Linux AS, for example, scales to eight CPUs and 16GB RAM quite a way off the capabilities of WS2003's Datacenter Edition with its 32 processors and 64GB RAM. The motives behind the two products are different, though: with its high-end offerings Red Hat is attempting to lure the customer who can't live without "proper support" into adopting an Open Source offering simply by offering that support; Microsoft is trying to make inroads into the massive-processing-power markets traditionally occupied by the IBM mainframes and the vast HP/Sun Unix machines.

Techworld's conclusion

If you're using anything but an Intel processor, Linux is the only choice. If your organisation is huge and you need what WS2003 Datacenter Edition gives you, you probably aren't reading this anyway. This aside, there's really nothing to choose between Windows and Linux with regard to the longevity of each incarnation; admittedly Microsoft keeps "mainstream" support on a product for much longer than the Linux companies, but the efforts of third parties (who deposit updated versions of applications for old versions of Linux on the Internet regularly) help negate this.

5. Migration

Which operating system you choose depends largely on what you want to do with it. If you have an Alpha-based server then the choice is simple you install Linux or you chuck it in the bin. But you're in a tiny minority if you own Alpha kit. Assuming we have a level playing field of 80x86-based equipment, the first step is to check the hardware compatibility list of the distribution you're thinking of installing, to make sure the specific equipment you have supports that incarnation of Linux. Then you have a few fundamental questions to ask yourself:

- Do I want the server to be a domain or Active Directory controller?
- Do I want the server simply to be a file and print server for Windows clients?
- Do I want to use the server to roll out Remote Installation Services images and/or software updates based on Windows policies?

If you answered "yes" to any of the above, go and buy a copy of Windows Server 2003. Although Linux can do some of these tasks, particularly the file-and-print functionality, it's really not worth the hassle when you consider that Windows does it all out of the box without any hassle at all. If, on the other hand, you answered "no" to all of the above, you still have the choice.

Next, you have to ask what skills you have and whether you need to train staff in order to support either of the alternatives. Bear in mind that Windows servers, in their basic form, are relatively easy to set up for an untrained but competent Windows-speaking systems manager. If you are installing a system of any significant size you ought to be considering Microsoft training for the Windows option just as hard as you should consider Linux training for the Linux alternative.

Now you need to consider the purpose the machine(s) will be put to. If your organisation is wedded to Exchange, or SQL Server, you're stuck with Windows. If you're open to options then you can consider Linux, because not only will commercial packages such as Oracle or DB2 fly like the wind, you also have choices, such as MySQL and PostgreSQL, which are free and efficient enough to support the majority of averagely-loaded database installations.

This leads neatly to the final question you need to ask yourself: will the savings I make on the software offset the increased cost of supporting a Linux installation? It's very easy to save several thousand pounds on software costs, particularly if you're running a database server, mail server, file-and-print server and all of your backups using Open Source software under Linux instead of paying for server copies and client access licences for commercial software under Windows.

But remember that total cost of ownership matters as well. If you have a tribe of MCSEs who would have to retrain to Linux, then there's a significant cost there. If, on the other hand, you have a set of techies whose knowledge would need to be enhanced just as much in a large new Windows installation as with a Linux setup, there's every chance that it'll be cheaper in the long run to move to Linux.

Migration instead of greenfield

What about migrating from one operating system to another? Well, the choice of what to migrate to includes all of the issues of selecting what to use in your from-the-ground-up installation, but with some important extras.

First is the skills issue. If you are already using systems in your organisation, then unless you've outsourced your support tasks, by definition you have experience in the existing systems. Regardless of whether you have formal training in whatever you use (in many organisations the techies have picked stuff up themselves on the job), you'll know many of the idiosyncrasies of the particular applications you use. You'll know that a particular error message is usually caused by disk X filling up, or by a user entering a wrong password, or by a failure of a disk in a particular array. By migrating you'll effectively have to start figuring out what's wrong from scratch again.

Then there's the application issue. Can you find an equivalent application for your existing stuff on the other platform? Does it do everything you need it to do? (Key question at this point; what matters is what you need it to do, not what it is capable of doing: how many of us have bought applications on the basis that it has 50 specific features and then only used half a dozen of them?)

Then comes the data migration issue. If you can find the right applications, how much of a horror story will it be when you come to move your system and user data to a different application on a different platform? Some data simply won't be movable without some kind of upheaval for the users (for instance, because both Linux and Windows use strong encryption algorithms, you can't decode the users passwords for transfer to the other system, although one choice is to continue with the existing directory service and just move the applications to a new platform).

When you're considering migrating from one platform to the other, there are a number of extra issues to consider on top of the decisions you'd make if installing from scratch. In short, the question to ask yourself is: "Does the benefit I'll gain offset the hassle of making the change?".

Techworld conclusion

If you absolutely have to go with a Microsoft-specific application, then that's what you have to do. This aside, Linux will always look like a better option from a cost point of view, but you have to ask the questions about the training requirements. If you're migrating rather than installing from scratch, there's an additional amount of effort you'll have to expend and the worry that your experience of the old system will go to waste, so you need to be confident that it's worth the bother.

6. Compatibility and interoperability

Although much of the time we've discussed Windows and Linux as alternatives, we have alluded in places to the idea of a heterogeneous environment where each tools is used to solve the issue(s) to which it's best suited, for instance running Windows-based ASP Web pages which attach to a Linux-based Oracle database at the back end. We should, therefore, look at the ways in which Windows and Linux can interoperate.

File sharing

Both Windows and Linux are able to share files in a number of ways. Windows' native mechanism uses the Server Message Block (SMB), which can be implemented on Linux via the Samba **(http://www.samba.org)** software package. Samba can act as both a server and a client so it can read and write from and to a remote Windows fileserver as well as acting as a server for Windows clients. Samba's relatively easy to configure, and it works with both domains and workgroups in a Windows environment.

For users who support Macintosh clients on their networks, there are a couple of alternatives, regardless of whether Linux or Windows is chosen as the server. One is to provide Mac-compatible fileserver volumes, which under Windows is done simply by installing Services for Macintosh, and which under Windows is provided by the Netatalk AppleTalk/AppleShare package **(http://rsug.itd.umich.edu/software/netatalk.html)**. An alternative is to configure the server as a Windows file sharing server and to run the Macs as Windows-compatible clients using a package such as DAVE **(http://www.thursby.com)**, which implements SMB at the client instead of requiring the server to speak AppleTalk.

Yet another way to proceed is to play to Linux's strengths and use the Network File System (NFS), which is the file sharing system inherent to the Unix family. This was the first approach used for connecting PCs to Unix machines back in the late 1980s and early 1990s, and there are several packages around that let you implement the NFS protocol on Windows (examples are AccessNFS - **http://www.accessnfs.com/nfs/** - and DiskAccess NFS Client - **http://www.simtel.net/product.php?url_fb_product_page=57907**). Oh, and you mustn't forget Windows' own Services for Unix, which comes on the Windows Server installation CD! On the Macintosh, life is very easy if you're using MacOS X you just connect and that's it (details at **http://www.withay.com/macosx/nfs-client.html**); with earlier versions of MacOS you,re unlikely to get very far, though, as NFS support for the Mac seems to have died out among the network software makers.

Printers and other services

Printer sharing is tightly coupled with file sharing, and so you often find that the same package that deals with the filesharing aspect handles the printing too. So if you're using Samba to share your Linux files with PCs and DAVE-enabled Macs, there's a section in the Samba configuration file where you can configure your printers too. Similarly, the Netatalk AppleShare service for Linux, and Windows' Services for Macintosh will deal with making the printers on the server available to Mac clients.

If you've chosen to use the Linux/Unix NFS filesharing approach, you have two choices on the client computer. The traditional approach would have been to get an application that made the PC into an LPD (the Unix printing standard) client. These days, however, LPD has been largely superseded by the Internet Printing Protocol (IPP), which is supported out of the box by most Linuxes and newer versions of Windows (notably WS2003 and Windows XP). If you're taking multi-platforms to extremes you could even run a Windows server to which your printers are connected as an LPD server **(http://lpd.brooksnet.com/)** so Linux machines can talk natively to them.

The main issue with printing to/from Linux servers is, as we've mentioned, one of driver availability. However, the driver issue is more relevant to the client (the machine actually doing the printing) than to the server that's hosting the printer. So if your Linux machine hosts a bunch of print queues for a clump of Windows clients, it's the clients that need the printer driver; drivers only really become an issue on the server with some of the modern multi-function printers, where the machine the printer is connected to is a scanner and fax machine, and can notify the computer when a document is being scanned or a fax comes in. By their nature, such devices tend to be connected to users' own PCs. Anyway, shared network printers tend to be reasonably fast laser printers that do nothing but print.

Directory services

Integration with directory services is a key requirement for a manageable network. If you want to run Active Directory then you'll certainly want a Windows server to act as the domain/AD controller, as it's just not realistic to do this under Linux. There's no problem with making a Linux server integrate into the Windows-driven AD structure and there are plenty of hand-holding guides to how to do it on the Net (including **http://jaxen.ratisle.net/~jj/nss_ldap-AD_Integration_how-to.html).**

You could, of course, decide to use Linux's native Network Information Services (NIS) structure, which continues to be supported by Microsoft with the Windows server family via the Services for Unix **(http://www.microsoft.com/windows/sfu/docs/sfuwp.doc)** add-on module. This is on the installer CD but isn't turned on unless you specifically ask for it, rather like Services for Macintosh.

Life is pretty sweet if you have a directory service that isn't native to Windows or Linux, most commonly Novell's eDirectory **(http://www.novell.com/products/edirectory/).** Novell provides client software for a variety of platforms, including both Windows and Linux, making it easy to run an eDirectory (formerly NDS) world from a master NetWare box but containing a bunch of different systems.

Application integration

One of the key failings with Linux server systems is the old chestnut that lack of software support is blocking its acceptance as a desktop operating system. If you're running a Windows-based fileserver, for instance, you probably have some kind of anti-virus system running on it, watching files as they arrive and checking for viruses. Because none of the popular corporate AV packages are available for Linux, though, you simply can't run mainstream anti-virus software on a Linux server although there are some products, from the likes of BitDefender **(http://www.bitdefender.com/bd/site/page.php)** that are beginning to address issues of virus protection on Linux machines.

On the other hand, there is a bonus with running Linux machines as servers – they're completely immune to Windows' viruses, which form the vast majority of attacks going around the Internet right now. Nobody's really caught on to the idea of writing Linux viruses yet (though Unix and its peers have had a few insecurity horror stories over the years, these have generally been bug exploitations, not viruses) and so what better operating system to be running on (say) the mail and file servers than one that's immune to MS Blaster and all those nasties we're hearing about on the TV.

Techworld's conclusion

Integration is something that both Windows and Linux are superb at both talking to each other and communicating with other platforms. There are dozens of potential combinations, and each seems to have at least one solution either in a commercial product (e.g. DAVE, the Mac-to-SMB package) or at no cost (e.g. Samba, the Linux-as-a-Windows-server package).

7. Making judgments

We set out to write this report with no preconceptions of which operating system would "win", partly because it's important to be independent but mainly because no two installations are the same and it's impossible to say "X is better than Y" in the general case. What has become clear, though, is that because Windows is far better supported by hardware and software vendors, there is a two-stage process.

At stage one, the bias is toward Windows, as you have to ask: "Do I want to do something that physically won't run on one of these operating systems?" It will generally be Linux that falls at this hurdle, because you'll either be trying to use hardware that Linux doesn't support, or you'll want to run a Windows-only application, or you'll want an Active Directory configuration.

At stage two, however, the bias is toward Linux, because you can now ask yourself: "Which is going to be the better value option". Aside from the potential training issues if you're historically a Microsoft house, the software costs will in the worst case (e.g. if you're using Oracle or some other commercial packages) be the same on both platforms. In the average case it will be lower if you choose Linux.

All of which seems to bring us to the conclusion:

- **Run a Windows server if Linux can't run the programs you want, on the server you want to use, in the authentication infrastructure you want.**
- **If you don't absolutely have to run Windows (for one of the above reasons) then unless the training requirements are extravagant, Linux will usually be the better value choice.**

Openness

The line that is always trotted out by the Open Source brigade in favour of choosing their approach instead of the commercial one is that if you can see the source code, you can examine it and see how it works. So if you're a top-secret government institution whose policy is that code needs not just to be tested thoroughly but to be proved correct mathematically, Linux and the Open Source regime are the option for you. On a smaller scale, with Open Source you have a copy of the source code and can re-compile or hack new features into it (an option that you don't have with commercial products) to provide some reassurance should the vendor go bust.

Unfortunately, the Open Source arguments don't work particularly well in a Linux vs. Windows discussion, because (a) if you're a big government, Microsoft will indeed let you see their source code; and (b) it's unlikely that Microsoft will go bust and leave you wondering where the next Windows patch is coming from. One has to wonder just how many people who opt for an Open Source solution because they get to see the code actually take the time to have a peek. You're talking hundreds of thousands of lines of semi-comprehensible C++ (for Linux, anyhow) and it's not for the faint-hearted.

In Open Source's favour is the fact that much of the code that's produced is better than the stuff it's replacing. Take the GNU software project, for instance, which over the years has come out with enormously complicated and useful packages like the Emacs text editor and the GCC compiler package. When I first installed GCC on the Sun-3 kit I was managing, it wasn't in the interests of promoting Open Source software, but because it ran faster and optimised code more efficiently than Sun's own commercial C compiler. In fact the installation was in two stages. First, you compiled GCC using the built-in Sun compiler, and then you used your new GCC compiler to recompile the GCC code because it made it go faster.

So although the principles of Open Source are often used as a selling point, remember that as it combines time and effort from a vast number of people, it generally evolves much more quickly than commercial software and is often better than its commercial equivalent. Okay, you get the downside that there's rarely any formal support for Open Source software (you can't call Emacs support, for instance) but it's fair to say that the mainstream Open Source stuff doesn't, in general, have any more bugs than the commercial offerings.

Pre-release offerings

In both the Open Source and commercial worlds, it's often possible to get beta-test, or sometimes even alpha-test versions of software. If you're a Microsoft registered developer, there's a world of pre-release software out there, and whenever an Open Source/Linux developer comes out with something new, it's not unusual to see alpha- and beta-test versions appearing on websites, with "Please let me know if it breaks" requests attached. The Apache Web server's Windows version was actually in alpha test right through the existence of version 1.x (i.e. for a number of years), and was stable enough for some companies to use it on their production Web hosts. The availability of pre-release software is essential for the system manager (and the business manager, for that matter) to know what's coming.

Usability

Usability is a small word used to describe a vast concept. The perception is that Linux is "not as usable" as Windows, server incarnations, because the former is inherently based on a command line and Windows has always had its GUI.

This isn't necessarily so, though. While it's true that Windows has a graphical interface, this doesn't make the concepts behind it any easier to understand. We can split usability issues into three rough sections.

The user interface

It's not unfair to say that, assuming you're running the X-Windows system on a Linux machine, the differences between the user interfaces of Linux and Windows are negligible. They all have the same concepts (windows, icons, menus, pointers) and although some of the programs and facilities are called different things, it's a simple job to get used to where they sit in the various menus and folders.

The configuration tools

It's true that Linux tends to err on the side of functionality, while Windows tends toward making everything graphical and wizard-based. These approaches appeal to different people: the wizard-based approach is great for newcomers but becomes tedious when you're used to the system; the functional approach is more cryptic for a start but makes life quicker when you've got the hang of things.

Take for example configuring a domain name server (DNS) on Linux and Windows. Windows' system is nicely graphical. It tells you when you've made a mistake and hides the nasty back-end DNS data structures from you. With Linux, you often end up editing a hideously scary text file by hand but if you want to start pointing several zones at a single file, or you have a number of zones that all have similar configurations, you can clone files and link files together with just a few keystrokes.

The concepts

In both systems, if you don't understand the concepts, you'll set something up wrongly. Regardless of whether you're configuring (say) a DHCP server under Windows or Linux, and regardless of whether you have a nice Windows interface or a text file to go at, if you don't understand what a network mask or a domain name is, you're going to do it incorrectly. Windows and Linux are unforgiving in different ways: Windows will lure you into a false sense of security and will do its best to tell you immediately where you've got the syntax wrong. Linux will only tell you you've done it wrong when you've built the whole configuration file and tried to start the server, but it'll tell you copiously in the logs.

The installer

These three classes of usability issue relate to ongoing maintenance, but what about the installer? Although different Linuxes have different installers (Mandrake is reckoned to be the best for beginners, though Red Hat has caught up a great deal) there really isn't much between Linux and Windows with regard to the installer. What actually catches most people out is the concepts behind the questions that those wizards ask. If you don't understand the concepts, then it doesn't matter a damn how friendly the installer is, you'll still manage to configure something incorrectly.

The pros and cons of Linux distributions

DISTRIBUTION	PRO	CON
Red Hat	Best supported by software vendors Has user confidence as a stable company that provides support.	Auto-updater is a paid-for service if you have more than one server
SuSE	Big, Teutonic and reliable	Has made some weird choices over the years, such as ReiserFS as the preferred file system type Not available for free as CD images.
Debian	Auto-updater works a treat, and there's no charge Supports loads of processors, not just Intel	Installer is traditionally more cryptic than the rest
Mandrake	Widely regarded as the easiest to install and use The best candidate for home and small office, non-expert, desktop users	Regarded more as a desktop Linux, through enterprise versions are available
Caldera	Probably the best installer we've come across	Lacking in popular support
SlackWare	One of the longest established distributions	Not commercially supported in the way that, say, Red Hat and SuSE are Installer isn't as good as Red Hat or Caldera

Techworld's conclusion

Of these six, which would we choose as a mainstream enterprise server platform? It would have to be between Debian and Red Hat. Debian has the benefit that "it just works" and you don't have to pay a bean for either the software itself or the update facility. Although driver support is generally Red Hat orientated, the restricting factors are usually related to the kernel, and you can update this yourself regardless of what distribution you're using. So, for example, although we once had a Debian-based server and a Red Hat driver for an Intel RAID card, we managed to get the driver working a treat with a bit of fiddling. So if you're not frightened to fiddle with your drivers, get Debian. If you want it idiot proof and you can afford a few tens of dollars a year, get Red Hat.

The heroes and villains of the vendor community

We arrived at the VHS (Vendor Hero Score) simply by looking at each vendor's website for Linux versions of their software and drivers, or comments to the effect of "coming soon". Note that this score is solely about the vendor's efforts to support Linux and disregards the plethora of third party drivers available for those vendor's products in the Linux community.

Vendor	Comments	Hero Score (out of 5)
IBM	IBM has consistently been one of Linux's strongest supporters among the leading hardware vendors - as shown by the launch of the first Tv ads for the operating system.	5
Cisco	Linux software for many products, including wireless LAN cards, VPN client software, Cisco SIP proxy server, Cisco Cable Diagnostic Manager, and iSCSI.	5
Intel	Drivers for many Ethernet adaptors, Linux driver CD ships with some RAID cards, drivers for some (but not all) modems.	3
3Com	Entering Linux in the Search box gave just two results. Linux config utility for Ethernet Client Bridge and the ASIC they sell OEM Ethernet device manufacturers. Linux drivers for many NICs and a small number of routers/DSL devices, but with unsupported plastered over most of the information sections.	1
NetGear	Drivers for some devices, including LAN and wireless LAN adaptors, but many devices are Windows-only.	3
HP/Compaq	Although not so good on the desktop, HP has embraced Linux on enterprise systems servers (including clustering), OpenView management software, firewall/AV products, etc.	4
Dell	Although we seem to remember Dell offereng Linux as a pre-installed server operating system, and in fact there are plenty of Linux-related downloads for Dell servers, when we looked (30 Aug 03) on their online store we were offered only Windows.	2

Key:

0	*Linux? What's that?*
1	*"We're working on our first Linux package"*
2	*Linux software available for one or more products*
3	*Linux software available for the most popular products*
4	*Linux software available for a good range of products*
5	*Windows? Nah, you should be using Linux*

About Techworld

Techworld is a website that provides news, reviews, analysis, how-to articles and technical White Papers to Britain's techie community. Written mainly by IT managers themselves, it's a fast-changing knowledge source that provides all the information that a technician needs to run a corporate network.

Built around a range of Knowledge Centres (of which, storage and security are the first) Techworld is constantly evolving and is set to become the first UK site that will offer highly relevant technical information on all aspects of enterprise networking. IDG

CONTACTS
EDITORIAL: Maxwell Cooter
(020 7071 3639)
editor@techworld.com

SALES: Guy Eaton
(020 7071 3610)
publisher@techworld.com

CIRCULATION: Jim Birch
(020 7071 3660)
circulation@techworld.com