Switching to free software—system administrators

An introductory guide for system administrators who want to switch to free software servers

Edward Macnaghten

You are a system administrator for a small company—the captain of the firm's computers. Doing your job well means that you may sail through the seas of information technologies unhindered, in short, the company's IT infrastructure will stay in place. Should you mess up you will find that the email has stopped working, the web surfers are stranded and you have pinned your ship on the reefs and rocks that scatter the virtual world, or in other words, the company will not be functioning well and you be burning its money.

I will imagine that you are running some Microsoft Windows server solutions. You may have some GNU/Linux boxes here and there for the less important tasks but, in essence, yours is a Microsoft house.

Then it finally happens. The CEO has received one too many viruses on the system, or he has just read the latest Microsoft licensing terms. He has decided to bite the bullet. The company is to move over to Linux servers, and it's you who are going to do it.

There is little, if anything, a Samba server cannot do that a Microsoft server can

But before I record the number butterflies in your stomach at this point, or that exciting feeling you experience as you sail into, what to your company is, virgin waters, I would like to borrow a time machine and go back to a point at which an important question was asked. It is, I think, if not the most, one of the most important questions there is in IT—and that is: *why*?

The reason why? Free software migration strategy

There is no real place for dogma in system administration. If a system works for you, then that is the system to use. Also, everything has a cost, and often the smallest cost is for software licenses. This is not usually high enough to appear on the radar of the CFO—with the possible exception of the larger corporations. Companies who have moved to free software are not usually concerned about licensing costs, but in freedom. When you have a free software solution, you have control. Should your software vendor decide to double the support fees, or you simply fall out, you can simply switch vendors without switching systems. Should your software provider go broke, you don't have to care. There is no problem getting the rights, or the source, to the software for others to support, and there would probably be no problem getting people to support it. An option that is rare in the proprietary world.

Using free software also means that you are no longer dependent on any single software vendor, or the good behaviour of one. Many proprietary companies hate the concept of free software because it relieves their customers' dependence on them. Although this is bad for the software company due to the fact they would have to provide good service in order to keep their revenue, it can't honestly be said that it's not good for the customer.

Your investment is also more secure with free software than it is with proprietary systems. Free software never becomes obsolete. Once it is written it stays written—should the original author or company decide to dump that product, it can still be picked up by someone else. An example of this is the "RedHat GNU/Linux" products. With the advent of their enterprise suite and the community Fedora products they decided to drop support for their old RedHat retail product. However, others picked up the baton and you can still get good support for these products from sources other than RedHat. The same can't be said about the old Windows products Microsoft has stopped supporting.

> It's a mistake to believe that there is no commercial level support for free software products

The strategy of any system deployment depends upon many things and is usually unique. The usual reasons for including a free software philosophy is not so much to reduce IT costs—which it tends to do anyway—but to increase your independence from individual IT vendors, protection of investment and freedom of choice. That's not counting the fact you'd also have access to the source code, and could change the program to suit your own needs should your requirements warrant it. This is a concept totally alien to proprietary software.

Now that the *why* is answered, the next question is *what*.

What to do? Analysis of free software migration

There are many IT server solutions, but I will be limiting the scope of this article to the assumption that you are an administrator of a Microsoft/IIS server deployment and you are migrating to a GNU/Linux one. However, This does not mean I am dismissing other free solutions such as OpenBSD (http://www.openbsd.org).

The whats, and often the ifs, of switching to free software solutions largely depend upon what your requirements are. When comparing the two types of systems, not only can free software perform standard server functions as well as Microsoft's offerings, but it usually does them better, including Windows' own networking functions. It's when transferring specialist legacy software from proprietary to free software systems, that you get the biggest problems. This legacy software is the software you currently have running, and whose functionality you would like to maintain, but is not available on the new platform. This is the biggest migration headache to free software causes. There are though, three possible solutions to this.

Sailing on the seas of system administration—responsibilities indeed

The first, as my grandmother would say about television, is simply to do without it. After all, she survived perfectly well without computers and Star Wars. So, there's no reason—she would contend—to have such reliance on the adventures of the Skywalkers and these new-fangled computer machines. Unfortunately, this solution is only feasible when the functionality of that legacy piece of software is near it's shelf life, or if it simply no longer fits the task it was designed for. In other words, you can really only do with out it when it's near being dumped anyway. There are circumstances where doing without could be the only solution, but this solution taken too far can attract employment costs, and while this may not have been a problem for my grandmother, it is one for a modern business. It's also undesirable in so far as it defeats the purpose of having computers in the first place.

The second solution is to use some kind of "bridge" technology, such as an emulator like Wine (http://www.

winehq.org), or virtualization software such as QEMU (http://fabrice.bellard.free.fr/qemu/), or to have a machine running the old proprietary system in the corner. This is cumbersome to maintain, and can spoil the goals of migrating. These solutions should only be used in exceptional circumstances where the legacy application is small, non critical and not used very much.

The third solution is to rewrite the legacy software, have it rewritten or to find a new equivalent package that performs the same task, and then migrate the data from the old software to the new. This is, in my opinion, the best solution. However, this can be costly in programmer costs, software vendor costs or staff training costs. A large section of Microsoft's inappropriately named "Get The Fact's" campaign lingers on this and calls it, somewhat misleadingly, "interoperable costs". These costs though are not reoccurring, they are one off, and quite often the costs you are forced to pay in upgrading and retraining on your current proprietary packages are not much cheaper than this. So it is this solution that is the way of the Jedi and where the force leads; where possible, chosen this route should be.

> That is the wonder of free software, so much has been written, and once it has been it stays available to all for eternity

It is worth noting that free software packages, and those that run on free software operating systems, are encompassing a wider and wider set of functionality, procedures and environments. Java is cross platform, also Microsoft's Dot-Net framework is no stranger to Linux due to Ximian's Mono, and there are other instances of similar migrations. Like the disappearing bathwater down the plug hole, the functionality that Microsoft proprietary systems can perform that free software cannot is getting less and less.

Once the *why* and *what* has been solved, the next question is *how*.

How to do it? The design of free software migration

At this stage of the game you need to decide what exactly to install. This is not trivial, and you will need to read the documentation floating about the web for details how to implement. A good resource is the HOWTOs at the linux doc site (http://www.linuxdoc.org). This article does not try to replace this or other resources—doing so would be well beyond the scope of one article—however, I have tried to supply some pointers that may be helpful.

Which Distribution?

At the top end there is enterprise level Linux from the likes of RedHat (http://www.redhat.com), Novell/SuSE (http://www.novell.com/products/openenterpriseserver/) and Mandrivia (http://www.mandrivia.com), also obtainable from companies such as IBM and HP. For the smaller business there are the community driven products such as the lighter versions of the above, Fedora (http://fedora.redhat.com), Suse Professional (http://www.novell.com/products/linuxprofessional) and Mandrivia.org (http://www.mandrivalinux.com/), Debian (http://www.debian.org) or many others (http://www.linux.org/dist/index.html).

Using non-enterprise distributions would then mean obtaining support from local third party vendors or doing it yourself. It's a mistake to believe that there is no commercial level support for free software products. There is, and it is more than adequate to base a business's IT on it. Many of the software packages required, including the ones listed below, are part of the distribution, don't need to be downloaded separately and have on-board documentation. I have included links though for reference.

Which GUI?

Fooled you! Trick heading. As far as software goes the first thing to point out is that Linux servers do not need a GUI. I find this often comes as a surprise to Microsoft system administrators, because with MS systems a GUI has to be installed. It is not necessary with Linux. In fact a GUI can get in the way. To make it clear—DO NOT install an "X Server program" (as they are misleadingly called). In fact, as far as the machine itself is concerned, once installed it needs neither a monitor nor keyboard.

Administration

Administration of the machine is done through the network in a number of ways. The easiest of these is through a product called webmin (http://www.webmin.com/), an intuitive interface for accessing many aspects of the Linux system through a web browser on a client. There are other ways but webmin is my favourite.

To digress for a moment, the configuration of services is handled slightly differently on GNU/Linux than in Microsoft's offerings. There is no registry as such, which in my personal opinion is a great relief. Services are usually controlled through text files of varying formats, which can be difficult to learn. However, this has been made far simpler by programs like webmin, which maintains these files for you presenting an intuitive front end to the administrator. Webmin is a modular program that is continuously growing and more modules are being added all the time to manage new services, so this utility grows as the Linux system itself does.

The above being said, there is nothing stopping an administrator from rolling up his sleeves and manually altering these files himself or herself by other means such as SSH (http://www.openssh.org). SSH stands for "Secure SHell" and is a secure way of logging onto the Linux system from a client and presenting the administrator with a console style terminal where Linux commands can be entered. However SSH can also be used to run X programs (GUI's) on the server. No—this is not contradicting the advice I gave a couple of paragraphs ago about not installing a GUI. The nature of the network features of X itself, enables the program's GUI interface to exist on one machine, while the program runs on another machine that doesn't have the GUI on it. The details of this are unfortunately, beyond the scope of this article, however more details can be found at X.org (http://www.x.org). On a side note, if you are maintaining the Linux server using a Microsoft Windows client and need to use the X functionality you could do worse than install Cygwin (http://sources.redhat.com/cygwin). This will give you this functionality and is of course, free software and free.

Web Server

For a web server you probably need Apache (http://www.apache.org/). Should you be migrating from Microsoft then you probably would've been running IIS. If your web site is just serving "html" files then there's probably not a lot you need to do to migrate—simply transfer the tree. There may be a snag or two but they are usually easily solved. For example, the case sensitivity of files does not matter on MS/IIS but does on Linux/Apache, however there is an extension to Apache (mod_speling.c) that takes care of this.

Administration utilities eliminate the need of a monitor and keyboard on the server

While I recommend Apache, there are lots of other web servers, many for specific environments, such as Lighttpd (http://www.lighttpd.net/) or Thttpd (http://www.acme.com/software/thttpd/) for a lightweight solution, or Jigsaw (http://www.w3.org/Jigsaw/) for a Java implementation.

Web Server Scripting

If scripting is involved then more than likely you are using ASP on IIS. There are ASP extensions to Apache, though these are limited and they are unlikely to have all of the class objects that are used in the scripts. Also there is a fundamental difference in how the two products work. IIS is thread based, where Apache is process based, and the way in which session variables are passed from connection to connection differ. All in all a rewrite is probably required in this case.

The nature of the network features of X itself, enables the program's GUI interface to exist on one machine, while the program runs on another machine that doesn't have the GUI on it

PHP is supported on both IIS and Apache, and the difference between the base products is hidden under a common

API. I recommend that PHP is what ends up being used on the Linux machine, and that it is previously installed on the IIS machine so that the web pages can be transferred to the new language in a controlled manner. There will still need to be testing and minor alterations when the move to the Linux box occurs but it would not be the mind-boggling difficult task that it would have been otherwise.

Under this heading comes the ubiquitous Perl (http://www.perl.com/) and Python (http://www.python.org/). Both of these exist under GNU/Linux and Windows environments so porting from one to the other should not be a great task.

Databases

If a database is used, such as MS-SQL or MS-Access, then I recommend either PostgreSQL (http://www.postgresql.org) or MySQL (http://www.mysql.org) on the Linux machine depending on its function. If mainly reads and very few writes are performed then MySQL is the better of the two, whereas if significant concurrent updates and complex queries, stored procedures and triggers are needed then PostgreSQL is the one for you. If MS-Access is currently used as a very light database then one like SQLite (http://www.sqlite.org), might suit you better. All of these are available on Microsoft-IIS as well as Linux-Apache so migration of these can occur prior to “switch over day”.

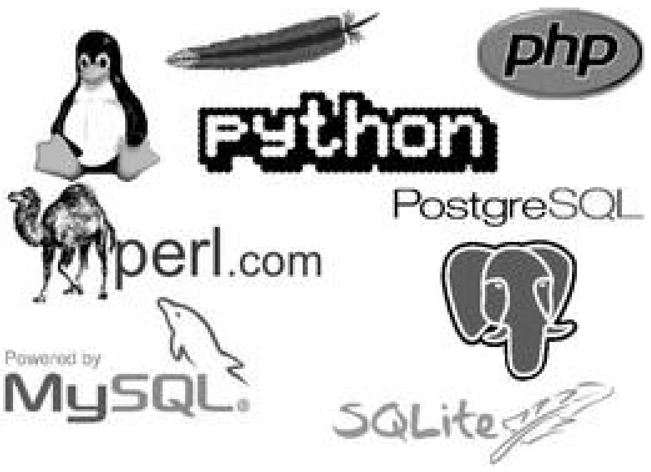
FTP

If FTP is required then Vsftpd (http://vsftpd.beasts.org/) is best. It is secure and highly configurable. Most distributions include this as standard. No more really needs to be said. For completeness sake however, I will mention there are many others, one of the more popular of these being the less secure Wu-ftp (http://www.wu-ftpd.org/).

Windows Networking

Although Linux servers are replacing Microsoft ones, it is less likely the same will be happening with the desktop machines. It is important that the servers supply the same role to these as before. Microsoft networking—CIFS as it is now known or NetBIOS as it used to be known—is taken care of by Samba (http://www.samba.org). There is little, if anything, a Samba server cannot do that a Microsoft server can. A “webmin” module can take care of a lot of this administration, and Samba itself has documentation on how to configure it. Once configured, it can provide a host of services including log-in domains, shares, printer queues and even roaming profiles.

The nuts and bolts of a Linux web server—all work on Windows too



Email Server

You are spoilt for choice with email servers and MTAs. There are four I come across all the time, so I will limit myself to them. They are Sendmail (http://www.sendmail.org/), Postfix (http://www.postfix.org/), Exim (http://www.exim.org/) and Qmail (http://www.qmail.org). Sendmail is the granddaddy of them all, though its configuration file can put people off. I lean toward using Exim and I’m happy with it, but I know others who feel the same way about Postfix and Qmail. Webmin modules exist for Sendmail, Qmail and Postfix, and Exim’s I have been informed is on its way.

Webmail

For webmail functionality again there is more than one product to choose from. I have used three of them in my lifetime—NeoMail (http://neocodesolutions.com/software/neomail/), Horde (http://www.horde.org/) and SquirrelMail (http://www.squirrelmail.org/). There are others. All three I’ve mentioned perform the function required of them, namely being able to view and manipulate mail on the server through a web browser.

Groupware

If you require Microsoft Exchange Group-ware functionality for interfacing with MS-Outlook then the most likely candidate that provides the major chunk of this functionality appears to be Open-XChange (http://www.open-xchange.org)—the free software cousin to Novell/SuSE’s proprietary OpenExchange product, however the MS-Outlook connectivity modules are proprietary extensions costing, at the time of writing, about US$20 a seat. This is something I have never needed so I cannot give my personal view on it, though reports suggest it works well, especially if you use the web interface. There are other web-based group-ware programs for Linux, such as PHPGroupware (http://www.phpgroupware.org) should MS-Outlook group-ware interaction not be required.

Spam Control

Spam filtering can be achieved with SpamAssassin (http://spamassassin.apache.org/), a Bayesian mail filtering program. This can be used to either flag email as spam by inserting a header, or by simply dumping it before it gets to the user.

Firewall and Proxy

Linux has its own firewall program called iptables, which comes with all distributions that I know of. This should be activated. Webmin has a configuration module for this, and often distributions come with one too. Configuring a firewall can be hard work at times, but doing so is a very good idea—regardless of operating system. Most Microsoft Windows machines that I have maintained relied upon a hardware router/firewall separate to the machine for its protection. While doing the same with Linux machines is not a bad idea either, it is not as critical. Experience dictates that the firewall software will keep the bad guys out more efficiently, but of course it does need to be configured correctly.

There is also Squid (http://www.squid-cache.org/). This is a highly versatile and configurable web and ftp web proxy-caching program. Issue 5 of Free Software Magazine contains an article Web site blocking techniques (http://www.freesoftwaremagazine.com/free_issues/issue_05/web_blocking_squid/) by Tedi Heriyanto, on Squid and its use.

Software Summary

There are literally thousands of other free software packages available for the Linux server:

- Mailman (http://www.gnu.org/software/mailman/) for managing mailing lists
- phpBB (http://www.phpbb.com/) for managing bulletin boards
- GeekLog (http://www.geeklog.net) for managing blogs
- Linux Terminal Server Project (http://www.ltsp.org/) for thin client solutions

Free software server applications—spoilt for choice

This list could go on and on, and doesn’t even scratch the surface. Should you require a specific piece of functionality you would probably find a quick Google search with a few well-chosen keywords will turn up trumps. However, you can also search sites like SourceForge (http://sourceforge.net/) or Freshmeat (http://freshmeat.net/) for an application to suit your requirements. That is the wonder of free software, so much has been written, and once it has been it stays available to all for eternity.

So, I’ve dealt with *why*, *how* and *what* have been catered for. Now for the crunch—w*hen*.

When to do it? The implementation of free software migration

It is at this point you will get those butterflies I mentioned earlier. You would have had the sense before now to have done dry runs of migrating of data and to have fully tested the new system. However, as with all of the other system implementations, things will crawl out the woodwork—or, in this case, computer-work—especially to make life annoying and difficult when the switch-over actually occurs.

The vast majority of those who have successfully made the change are grateful that they did so. Once done, they found they were no longer tied down and forced to upgrade at the whim of a vendor

If possible, it is advantageous to do one machine at a time. The Linux firewall iptables program will permit defined TCP ports to be redirected to old machines if required, so it's even possible to transfer one service at a time. Like so many other things in this game, there's no set of defined rules as to how to go. Just make sure you've cleared your desk of other tasks in order to keep yourself available to sort out problems for the first week or so.

> When comparing the two types of systems, not only can free software perform standard server functions as well as Microsoft's offerings, but it usually does them better, including Windows' own networking functions

Conclusion

Once the initial pain and expense has been suffered by making the transition then savings will usually start to show. The *real* facts are that the free software solutions are more secure, more stable and cheaper to run, even before counting the free licensing savings.

> Like the disappearing bathwater down the plug hole, the functionality that Microsoft proprietary systems can perform that free software cannot is getting less and less

The vast majority of those who have successfully made the change are grateful that they did so. Once done, they found they were no longer tied down and forced to upgrade at the whim of a vendor. Should you successfully migrate then you decide when you want to upgrade. You have the knowledge. You can grab the support back from your IT company, without fear that the product is no longer supported. You find yourself sailing on the clear and open ocean rather than being tied to inconvenient shipping lanes. You are plotting your own course. You are now free.

Feeling the freedom of free software

Copyright information

About the author

Edward Macnaghten has been a professional programmer, analyst and consultant for in excess of 20 years. His experiences include manufacturing commercially based software for a number of industries in a variety of technical environments in Europe, Asia and the USA. He is currently running an IT consultancy specialising in free software solutions, which is based in Cambridge, UK. He maintains a blog on his web site (http://eddy.edlsystems.com).