Migration and coexistence with non-Linux systems

Josep Jorba Esteve

PID_00148467

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Index

Introduction

Having had a brief introduction to GNU/Linux systems, the following step is to integrate them in the work environment as production systems. According to the current system in use, we can consider either a full migration to GNU/Linux systems or a coexistence through compatible services.

Migration to the GNU/Linux environment may be done progressively by replacing services partially or by substituting everything in the old system by GNU/Linux equivalents.

In current distributed environments, the most relevant concern is the client/server environments. Any task in the global system is managed by one or more dedicated servers, with the applications or users directly accessing the offered services.

Regarding the work environment, whether in the simplest case of the individual user or the more complex case of a business environment, every environment will require a set of services that we will need to select, later adjusting client and server machines so that they can access them or provide their use.

The services may encompass different aspects and there tend to be various types for sharing resources or information. File servers, print servers, web servers, name servers, e-mail servers etc., are common.

The administrator will normally select a set of services that need to be present in the work environment according to the needs of the end users and/or the organisation; and must configure the right support for the infrastructure, in the form of servers that support the expected workload.

1. Computer systems: environments

During the process of installing some GNU/Linux distributions, we often find that we are asked about the type of environment or tasks our system will be dedicated to, which often allows us to choose a sub-set of software that will be installed for us by default, because it is the most suited to the contemplated job. We will often be asked if the system will be used as a:

a) Workstation: this type of system usually incorporates particular applications that will be used most frequently. The system is basically dedicated to running these applications and a small set of network services.

b) Server: basically it integrates most network services or, in any case, a particular service, which will be the system's main service.

c) Dedicated calculation unit: calculation-intensive applications, renders, scientific applications, CAD graphics etc.

d) Graphics station: desktop with applications that require interaction with the user in graphic form.

We can normally set up our GNU/Linux system with one or more of these possibilities.

More generally, if we had to separate the work environments [Mor03] where a GNU/Linux system can be used, we could identify three main types of environments: workstation, server and desktop .

We could also include another type of systems, which we will generically call embedded devices or small mobile systems like a PDA, mobile telephone, portable video console etc. GNU/Linux also offers support for these devices, with smaller personalised kernels for them.

Note

GNU/Linux systems can be dedicated to server, workstation or desktop functions.

Example

For example, we should mention the initial work done by the Sharp company on its Zaurus models, a PDA with advanced Linux features (there are four or five models on the market). Or also other Linux initiatives of an embedded type such as POS (point of sale) terminals. Or video consoles such as GP2X, and Sony Playstation 3 linux support. Also new smartphone/PDA platforms like Google Android, Nokia Maemo, Intel Moblin.

Regarding the three main environments, let's look at how each one of these computer systems is developed in a GNU/Linux environment:

1) A workstation type system tends to be a high performance machine used for a specific task instead of a general set of tasks. The workstation classically consisted of a high performance machine with specific hardware suited to the task that needed doing; it was usually a Sun's SPARC, IBM's RISC or Silicon Graphics machine (among others) with its variants of proprietary UNIX. These high cost machines were oriented at a clear segment of applications, whether 3D graphic design (in the case of Silicon or Sun) or databases (IBM or Sun). Nowadays, the performance of many current PCs is comparable (although not equal) to these systems and the frontier between one of these systems and a PC is no longer clear, thanks to the existence of GNU/Linux as an alternative to the proprietary UNIX versions.

2) A server type system has a specific purpose, which is to offer services to other machines on the network: it offers a clearly distinct set of characteristics or functionality from other machines. In small computer systems (for example, with less than 10 machines), there is not usually an exclusive server system, and it tends to be shared with other functionalities, for example as a desktop type machine. Medium systems (a few dozen machines) tend to have one or more machines dedicated to acting as a server, whether as an exclusive machine that centralises all services (e-mail, web etc.) or as a pair of machines dedicated to sharing the main services.

In large systems (hundreds or even thousands of machines), the load makes it necessary to have a large group of servers, with each one usually exclusively dedicated to a particular service, or even with a set of machines exclusively dedicated to one service. Moreover, if these services are provided inwards or outwards of the organisation, through access by direct clients or open to the Internet, depending on the workload to be supported, we will have to resort to SMP multicore type solutions (machines with multiple processors/code) or of the cluster type (grouping of machines that distribute a particular service's load).

The services that we may need internally (or externally) can encompass (among others) the following service categories:

a) Applications: the server can run applications and as clients we just observe their execution and interact with them. For example, it may encompass terminals services and web-run applications.

b) Files: we are offered a shared and accessible space from any point of the network where we can store/recover our files.

c) Database: centralisation of data for consultation or production by the system's applications on the network (or for other services).

d) Printing: there are sets of printers and their queues and jobs sent to them from any point of the network are managed.

e) E-mail: offers services for receiving, sending or resending incoming or outgoing mail.

f) Web: server (or servers) belonging to the organisation for internal or external use by customers.

g) Network information: for large organisations it is vital for finding the services offered or the shared resources; or users themselves, if they need services that make this localisation possible and to consult the properties of each type of object.

h) Names services: services are required to name and translate the different names by which the same resource is known.

i) Remote access services: in the case of not having direct access, we need alternative methods that allow us to interact from the outside, giving us access to the system that we want.

j) Name generation services: in naming machines, for example, there may be a highly variable number of them, or they may not always be the same ones. We need to provide methods for clearly identifying them.

k) Internet access services: many organisations have no reasons for direct access and rather have access through gateways or proxies.

l) Filtering services: security measures for filtering incorrect information or information that affects our security.

3) A desktop type machine would simply be a machine used for routine everyday computer tasks (such as our home or office PC).

Example

For example, we could establish the following as common tasks (included in some of the most used GNU/Linux programs):

- Office tasks: providing the classical software of an office suite: word processor, spreadsheet, presentations, a small database etc. We can find suites like OpenOffice (free), StarOffice (paid for, produced by Sun), KOffice (by KDE), or various programs like Gnumeric, AbiWord which would form part of a suite for Gnome (known as Gnome-Office).

- Web browser: browsers such as Mozilla Firefox, Konqueror, Epiphany etc.

- Hardware support (USB, storage devices...). Supported in GNU/Linux by the appropriate drivers, usually provided in the kernel or by the manufacturers. There are also new hardware analysis tools such as kudzu (Fedora/Red Hat) or discover (Debian). Media and entertainment (graphics, image processing, digital photography, games and more). In GNU/Linux there is an enormous amount of these applications of a very professional quality: Gimp (touching up photographs), Sodipodi, Xine, Mplayer, gphoto etc.

- Connectivity (remote desktop access, access to other systems). In this regard, GNU/Linux has an enormous amount of own tools whether TCP/IP or FTP, telnet, web etc., or X Window, which has remote desktop capabilities for any UNIX machine, rdesktop (for connecting to Windows desktops), or VNC (for connecting to UNIX, Windows, Mac etc.).

Web sites

Open Source office suites:

http://openoffice.org
http://www.koffice.org/
http://live.gnome.org/Gnome-Office

2. GNU/Linux services

GNU/Linux has servers adapted for any work environment.

The service categories we have mentioned have equivalents that we can provide from our GNU/Linux systems to all other machines on the network (and from which they can also act as clients):

a) Applications: GNU/Linux can provide remote terminal services, whether by direct connection through series interfaces of dumb terminals, serving to visualise or interact with the applications. Another possibility is remote connection in text mode, from another machine via TCP/IP services such as rlogin, telnet, or in a secure way with ssh. GNU/Linux provides servers for all these protocols. In the case of running graphics applications, we have remote solutions through X Window, any UNIX, Linux or Windows client (or others) with an X Window client can visualise the running of the environment and its applications. At the same time, there are other solutions such as VNC for the same problem. Regarding the issue of web-run applications, GNU/Linux has the Apache server, and any of the multiple web running systems are available, whether Servlets (with Tomcat), JSP, Perl, PHP, xml, webservices etc., as well as web application servers such as BEA Weblogic, IBM Websphere, JBoss (free) which are also run on GNU/Linux platforms.

b) Files: files can be served in various ways, either through FTP access to the files, or by serving them in a transparent manner to UNIX and Linux machines with NFS, or by acting as client or server towards Windows machines through Samba.

c) Database: it supports a large number of relational client/server type databases such as MySQL, PostgreSQL and several commercial ones such as Oracle or IBM DB2, among others.

d) Printing: it can serve local or remote printers, for both UNIX systems with TCP/IP protocols and Windows through Samba/CIFS.

e) E-mail: it offers services for clients to obtain mail on their machines (POP3 or IMAP servers), as mail transfer agents (MTA) to recover and retransmit mail, such as the Sendmail server (UNIX standard) or others like Exim and, in the case of outward sending, the SMTP service for outgoing mail.

f) Web: we have the http Apache server, whether in its 1.3.x versions or the new 2.0.x. or 2.2.x. versions Also, we can integrate web application servers, such as Tomcat for servlets, JSP...

g) Network information: services such as NIS, NIS+ or LDAP allow us to centralise the information from the machines, users, and various resources on our network, facilitating administration and service to users, in such a way that the latter do not depend on their situation in the network. Or if our organisation has a certain internal structure, these services will allow us to model it allowing access to the resources to whoever needs it.

h) Names services: services such as DNS for machine names and their translation from or to IP, by means of the Bind server for example (the standard UNIX DNS).

i) Remote access services: whether to run applications or to obtain remote information on the machines. The servers could be the ones we have mentioned for the applications: X Window, VNC etc., and also those that allow some remote commands to be run without interactivity such as rexec, rsh, ssh etc.

j) Name generation services: services such as DHCP allow TCP/IP networks, to dynamically (or statically) generate the available IP addresses according to the machines that need it.

k) Internet access services: in certain situations there may be a single output to Internet (or several). These points tend to act as proxy, since they have access and they redirect it to potential Internet accesses on behalf of clients. They also tend to act as content cache. In GNU/Linux we can have Squid for example. In this category, a gateway or router could also come into action in a GNU/Linux system, whether to direct packages to other networks or to find alternative resending routes. Also, in the case of small installations such as domestic ones, we could include the Internet access by modem through the PPP services.

l) Filtering services: one of the most commonly used security measures at present is firewalls. They basically represent filtering techniques for incoming or outgoing packages, for the different protocols we are using, to put up barriers against unwanted ones. In GNU/Linux, we have mechanisms such as ipchains and iptables (more modern) for implementing firewalls.

3. Types of use

GNU/Linux, as a system, offers characteristics that are valid for personal users as well as users of a medium or large-scale infrastructure.

From the perspective of GNU/Linux system users, we could distinguish:

a) The individual or domestic user: normally, this type of user has one or several machines at home that may or may not be shared. In general, in this environment, GNU/Linux is used to develop a desktop system, which means that the graphics part will be important: the GNU/Linux desktop.
For this desktop we have two main options in the form of Gnome and KDE environments, both of which are perfectly valid. Either of the two environments offers applications running and visualisation services, together with a broad range of basic own applications that allow us to develop all sorts of routine tasks. The two environments offer a visual desktop with different menus, icon bars and icons, in addition to navigators for own files and various useful applications. Any environment can run its own applications and the others', although, in the same way as the applications, they run better in their own environment because their visual aspect is more suited to the environment for which they were designed.
Regarding applications for the personal user, we should include the typical ones of the desktop system. If the user has a home network, for example, a small group of computers joined by an Ethernet type network, services for sharing files and printers between machines could also be interesting. Services such as NFS may be necessary if there are other Linux machines; or Samba, if there are machines with Windows.
In the case of having an Internet connection through an ISP (Internet Service Provider) depending on the type of connection used, we would need to control the corresponding devices and protocols:

- Modem connection: telephone modems tend to use the PPP protocol to connect with the provider. We would have to enable this protocol and configure the accounts we have enabled with the provider. An important problem with Linux is the winModems issue, which has caused a lot of trouble. This modem (with some exceptions) is not supported, because it is not a real modem but rather a hardware simplification plus driver software, and most only function with Windows, meaning that we need to avoid them (if not supported) and to buy real (full) modems.

- ADSL modem connection: the functioning would be similar, the PPP protocol could be used or another one called EoPPP. This may depend

on the modem's manufacturer and on the type of modem: Ethernet or USB.

- ADSL connection with a router: the configuration is very simple, because in this situation all we need to do is to configure the Ethernet card and/or wireless card in our system to connect with the ADSL router.

Once the interface to Internet is connected and configured, the last point is to include the type of services that we will need. If we only want to act as clients on Internet, it will be sufficient to use the client tools of the different protocols, whether FTP, telnet, the web navigator, e-mail or news reader etc. If we also wish to offer outgoing services – for example, to publish a website (web server) or to allow our external access to the machine (ssh, telnet, FTP, X Window, VNC, services etc.), in this case, server – then we must remember that this will only be possible if our provider gives us fixed IP addresses for our machine. Otherwise, our IP address will change every time we connect and the possibility of offering a service will become either very difficult or impossible.

Another interesting service would be sharing access to the Internet between our available machines.

b) Mid-scale user: this is the user of a middle scale organisation, whether a small company or group of users. Normally, this type of users will have local network connectivity (through a LAN, for example) with some connected machines and printers. And will have direct access to Internet, either through some proxy (point or machine designed for an external connection), or there will be a few machines physically connected to the Internet. In general, in this environment, work is partly local and partly shared (whether resources, printers or applications). Normally, we will need desktop systems; for example, in an office we can use office suite applications together with Internet clients; and perhaps also workstation type systems; for example, for engineering or scientific jobs, CAD or image processing applications may be used, as well as intensive mathematical calculation systems etc., and almost certainly more powerful machines will be assigned to these tasks.

In this user environment, we will often have to share resources such as files, printers, possibly applications etc. Therefore, in a GNU/Linux system, NFS services will be appropriate, printer services, Samba (if there are Windows machines with which files or printers need to be shared), and we may also need database environments, an internal web server with shared applications etc.

c) Large-scale users: this type of user resembles the preceding one and differs only in the size of the organisation and available resources, which can be plenty, in such a way that some resources of the NIS, NIS+ or LDAP type network system directory may be needed in order to handle the organisation's information and reflect its structure, certainly also to

have large service infrastructures for external clients generally in the form of websites with various applications.

This type of organisation has high levels of heterogeneity in both system hardware and software, and we could find lots of architectures and different operating systems, meaning that the main tasks will consist of easing data compatibility by means of databases and standard document formats and to ease interconnectivity by means of standard protocols, clients and servers (usually with TCP/IP elements).

4. Migration or coexistence

Next, we will consider another important aspect in adopting GNU/Linux systems. Let's suppose that we are amateurs at handling this system; or, the opposite, that we are experienced and wish to adopt one or several GNU/Linux systems as individual users for working in our small organisation; or that we are considering replacing the infrastructure of our large company or organisation in full (or part).

Migrating to a new system is no trivial matter, it needs to be evaluated through a study that analyses both the costs and the beneficial features that we expect to obtain. Also, migration can be done in full or in part, with a certain degree of coexistence with former systems.

We will be dealing with a full or partial migration project of our IT systems to GNU/Linux and, as administrators, we will be responsible for this process.

As in any project, we will have to study the way of responding to questions such as: Does the change make sense in financial terms or in terms of performance benefits? What is the migration's objective? What requirements will we want to or need to fulfil? Can we do a partial migration or do we need to do a full migration? Is coexistence with other systems necessary? Will we need to retrain users? Will we be able to use the same hardware or will we need new hardware? Will there be important added costs? Or simply, will it go okay? These and many others are the questions that we will have to try and answer. In the case of a company, the answers would be provided in a migration project, specifying its objectives, requirements, the implementation process, and including a financial analysis, user training plans etc. We will not go into this in detail, but will consider some of these issues in a simple manner. And in the final workshop we will examine a few small cases of how we would implement the migration.

Also, the moment we start migrating to GNU/Linux, we will start to notice the advantages the system brings to our organisation:

a) Costs: reduction in license costs for the system's software and applications. GNU/Linux has 0 cost for licenses if purchased from the Internet (for example, in the form of images from the distribution's CDs), or a negligible cost if we take into account that the nearest comparison for systems with equivalent features would be Windows Server systems with license costs ranging between € 1,500 and € 3,000, without including a large amount of the additional software that a typical GNU/Linux distribution would include.

But careful, we should not underestimate maintenance and training costs. If our organisation consists solely of users and administrators trained in Windows, we may have high costs for retraining personnel and, possibly, for maintenance. Therefore, many big companies prefer to depend on a commercial distributor of GNU/Linux to implement and maintain the system, such as the business versions offered by Red Hat, SuSe and others. These GNU/Linux versions also have high license costs (comparable to Windows), but at the same time are already adapted to business structures and contain their own software for managing companies' IT infrastructure. Another important aspect, to conclude with cost estimates, is the TCO concept (total cost of ownership), as a global evaluation of the associated costs that we will find when we undertake a technological development; we don't just have to evaluate the costs of licenses and machines, but also the costs of training and support for the people and products involved, which may be as high or more than the implemented solution.

b) Support: GNU/Linux offers the best maintenance support that any operating system has ever had, and it is mostly free. Nevertheless, some companies are reluctant to adopt GNU/Linux on the basis that there is no product support and prefer to buy commercial distributions that come with support and maintenance contracts. GNU/Linux has a well-established support community worldwide, through various organisations that provide free documentation (the famous HOWTOs), specialised user forums, communities of users in practically any region or country in the world etc. Any question or problem we have can be searched on the Internet and we can find answers within minutes. If we don't, if we have found a bug, error, or untested situation, we can report it on various sites (forums, development sites, distribution bug sites etc.), and obtain solutions within hours or, at the most, within days. Whenever we have a question or problem, we should first try a few procedures (this is how we will learn) and if we do not find the solution within a reasonable amount of time, we should consult the GNU/Linux community in case any other user (or group of users) has encountered the same problem and found a solution, and if not, we can always post a report on the problem and see if we are offered solutions.

Note

Linux Howto's: http://www.tldp.org/

4.1. Identify service requirements

Normally, if we have systems that are already functioning we will have to have some services implemented for users or for helping the infrastructure of the IT support. The services will fall within some of the categories seen above, with the GNU/Linux options that we mentioned.

GNU/Linux systems are not at all new, and as we saw in the introduction, stem from a history of more than thirty years of UNIX systems use and development. Therefore, one of the first things that we will find is that we are not lacking support for any type of service we want. If anything, there will

be differences in the way of doing things. Also, many of the services used by IT systems were conceived, researched, developed and implemented in their day for UNIX, and only subsequently adapted to others systems (such as Windows, more or less successfully).

Many companies with proprietary UNIX participate in GNU/Linux and offer some of their developments to the community.

Any service available at the time may be adapted to GNU/Linux systems with equivalent (if not the same) services.

Example

A famous case is the one of the Samba servers [Woo00] [Sam]. Windows offers what it calls "sharing files and printers on the network" by means of its own protocols known generically as SMB (server message block) [Smb] (with network support in the NetBios and NetBEUI protocols). The name CIFS (common Internet file system) is also commonly used, which is what the protocol was called in a second revision (which continued to include SMB as a basic protocol). These protocols allowed the sharing of files (or disks) and printers on a network of Windows machines (in a workgroup configuration or in Windows domains). In UNIX this idea was already old when it appeared in Windows and services such as NFS for sharing files or managing printers remotely were already available using TCP/IP protocols.

One of the problems with replacing the Windows sharing services based on NetBios/NetBeui (and ultimately with NetBios over TCP/IP) was how to support these protocols, since if we wanted to keep the client machines with Windows, we could not use the UNIX services. For this purpose, Samba was developed as a UNIX server that supported Windows protocols and that could replace a Windows server/client machine transparently, with client users with Windows not having to notice anything at all. Moreover, the result in most cases was that the performance was comparable if not better than in the original machine with Windows services.

Currently, Samba [Sam] is constantly evolving to maintain compatibility with Windows file and printer sharing services; because of the general changes that Microsoft subjects SMB/CIFS [Smb] protocols to (the base implemented by Samba) with each new Windows version, in particular the evolution of workgroup schemes in the operating systems' client versions, to centralised server (or group of servers) schemes, with specific user authentication services (NTLM, NTLMv2, Kerberos), and centralised storage of the system's management such as Active Directory. In addition to this, the configuration of existing domain servers (whether with primary controller, backup or Active Directory).

Currently, in migration processes with Samba, we will need to observe what configurations of Windows clients/servers (and its versions) exist on the system, as well as what user authentication and/or information management systems are used. Also, we will need to know how the system is structured into domains (and its controller servers, members or isolated servers), in order to make a complete and correct mapping towards Samba-based solutions, and into complementary user authentication (winbind, kerberos, nss_ldap) and management services (for example openLDAP) [Sama] [Samb] .

4.2. Migration process

In the migration process, we need to consider how we want to migrate and if we want to migrate totally or partially, coexisting with other services or equipment that has a different operating system .

In the environments of large organisations, where we find a large number of heterogeneous systems, we will need to take into account that we will almost certainly not migrate every one of them, especially workstation type systems

that are dedicated to running a basic application for a specific task; it could be that there is no equivalent application or simply that we wish to keep these systems for financial reasons or in order to maximise an investment.

We can migrate various elements, whether the services we offer, the machines that offer the services or the clients who access the services.

Elements that can be migrated include:

a) Services or machines dedicated to one or more services. In migrating, we will replace the service with another equivalent one, normally with minimum possible impact unless we also wish to replace the clients. In the case of Windows clients, we can use the Samba server to replace the file and printer services offered by the Windows machines. For other services, we can replace them with GNU/Linux equivalents. In the case of replacing just one service, normally we will disable the service on the machine that offered it and enable it on the new system. Client changes may be necessary (for example, new machine addresses or parameters related to the service).

If a server machine was responsible for an entire function, we will need to analyse whether the machine was dedicated to one or more services and whether they can all be replaced. If so, we will just have to replace the old machine with the new one (or maintain the old one) with the services under GNU/Linux and in any case, modify a client parameter if necessary. Normally, before making a change, it is advisable to test the machine separately with a few clients in order to make sure that it performs the function correctly and then to replace the machines during a period when the system is inactive.

In any case, we will certainly have to back up data existing prior to the new system, for example, file systems or the applications available in the original server. Another point to consider in advance is data portability; a problem we often find is compatibility when the organisation used data or applications that depended on a platform.

Example

To mention a few practical cases that some companies find nowadays:

- Web applications with ASP: these applications can only be executed on web platforms with Windows and Microsoft's IIS web server. We should avoid them if we intend to migrate platforms at any time and don't wish to rewrite them or pay another company to do so. GNU/ Linux platforms have the Apache web server (the most commonly used on the Internet), which can also be used with Windows, this server supports ASP in Perl (in Windows it generally uses visual basic, C# and Javascript), there are third party solutions to migrate ASP or to more or less convert them. But if our company depended on this, it would be very costly in terms of time and money. A practical solution would have been to make the web developments in Java (which is portable between platforms) or other solutions such as PHP. On this point, we should highlight the Mono project [Mon] (sponsored by Novell) for portability of part of Microsoft's .NET environment to GNU/Linux, in particular a large amount of the.NET API's, C# language, and the ASP.NET specification. Allowing a flexible

migration of .NET applications based on .NET APIs that are supported by the Mono platform. At the same time, we should mention the FSF's DotGnu [Dgn] project, as a GPL alternative to Mono.

- Databases: using a Microsoft SQL Server for example, makes us totally dependant on its Windows platform, plus, if we use proprietary solutions in a specific environment for database applications, they will be difficult to transfer. Other databases such as Oracle and DB2 (IBM) are more portable because they have a version in the different platforms or because they use more portable programming languages. We could also work with PostgreSQL or MySQL database systems (it also has a version for Windows) available in GNU/Linux, and that allow an easier transition. At the same time, if we combine it with a web development we have a lot of possibilities; in this sense, nowadays we use systems such as: web applications with Java, whether servlets, applets, or EJB; or solutions such as the famous LAMP, the combination of GNU/Linux, Apache, Mysql and Php.

b) Workstation: in these migrations, the biggest problem stems from the applications, whether for CAD, animation, engineering or scientific programs, which are the workstation's main reason for being. Here it will be important to be able to replace them with equal or at least compatible applications with the same expected features or functionality. Normally, most of these applications stem from a UNIX world, given that most of these workstations were conceived as UNIX machines. Meaning that a compilation or minimum adaptation to the new GNU/Linux may be enough, if we have source code (as tends to be the case with many scientific applications). If we are dealing with commercial applications, the manufacturers (of engineering and scientific software) are starting to adapt them to GNU/Linux, although in these cases the applications are usually very expensive (easily hundreds to thousands of euros).

Note

For examples of GNU/Linux equivalent applications, see:
http://www.linuxalt.com/
http://wiki.linuxquestions.org/wiki/Linux_software_equivalent_to_Windows_software
http://www.linuxrsp.ru/win-lin-soft/table-eng.htmlç

c) Desktop client machines. Desktop machines continue to be a headache for the world of GNU/Linux, because they involve a number of additional problems. In servers, the machines are assigned clear functionalities, as a rule they do not require complex graphic interfaces (often text communication is sufficient), and the normally specific high performance hardware is purchased for a specific set of functions and the applications tend to be the servers themselves included in the operating system or some third party applications. Also, these machines are often managed by administrators with extensive knowledge of what they are dealing with. However, in the case of desktops, we are dealing with a problem factor (in itself and more so for administrators): the system's end users. The users of desktop systems expect to have powerful graphic interfaces that are more or less intuitive and applications that allow them to run routine – usually office – tasks. This type of user (with a few exceptions) has no reason to have advanced knowledge of computers; in general, they are familiar with office suites and use a couple of applications with varying degrees of skill. Here GNU/Linux has a clear problem, because UNIX as such was never conceived as a purely desktop system and was only later adapted with graphic interfaces such as X Window and the different desktops,

such as the current GNU/Linux ones: Gnome and KDE. Furthermore, the end user tends to be familiar with Windows systems (which have almost a 95% share of the desktop market).

In the case of desktops, GNU/Linux has a number of obstacles to overcome. One of the most critical ones is that it does not come preinstalled on machines, which obliges the user to have a certain amount of knowledge in order to be able to install it. Other reasons could be:

Note

The desktop environment is a battle yet to be waged by GNU/Linux systems; which need to defeat users' reluctance to switch systems and generate awareness of their ability to offer simple alternatives and applications that can handle the tasks demanded by users.

• User reluctance: a question a user may ask is: Why should I switch system? Will the new environment offer me the same thing? One of the basic reasons for changing will be quality software and its cost, since a large proportion will be free. On this point, we should consider the issue of illegal software. Users seem to consider that their software is free, when really they are in an illegal situation. GNU/Linux software offers good quality at a low cost (or at no cost in many cases), with several alternatives for the same job.

• Simplicity: users are normally lost if the system does not have similar reference points to those the user is already familiar with, such as interface behaviour or tools with similar functionality. Users generally expect not to have to spend too much extra time on learning how to handle the new system. GNU/Linux still has a few problems with more or less automatic installations, which means that a certain amount of knowledge is still required in order to install it correctly. On this point, we should mention the ease of installing it in different environments provided by recent desktop oriented distributions like Ubuntu [Ubu]. Another common problem concerns support for the PC hardware; even though it is improving all the time, manufacturers still don't pay enough attention to it (partly for reasons of market share). Until there is a clear intention in this regard, we will not be able to have the same support as other proprietary systems (like Windows). However, we should emphasise the work of the Linux kernel community to offer the right support for new technologies, in some cases by supporting the manufacturer or by preparing primary support (if not supported by the manufacturer) or alternative support to that offered by the manufacturer.

• Transparency: GNU/Linux environments have many complex mechanisms, such as daemons, services, difficult to configure ASCII files etc. For end users, it should be necessary to hide all of these

complexities by means of graphics programs, configuration wizards etc. This is the path taken by some distributions such as Red Hat, Mandriva, Ubuntu or SuSe.

• Support for known applications: a standard office suite user will face the problem of data portability or handling data formats. What to do with existing data? This problem is being solved daily, thanks to the office suites that are starting to have the functionalities a desktop user needs. For example, if we consider a migration from using a Windows Office suite, we can find suites such as OpenOffice (free software) that can read (and create) the formats of Office files (with some restrictions). Format compatibility is not difficult when it is open, but in the case of Windows, Microsoft continues to maintain a policy of closed formats; and a serious amount of work is needed in order to be able to use these formats, by means of reverse engineering (a fairly costly process). Also, in the Internet age, when information is supposed to move about freely, undocumented closed formats are more an obstacle than anything else. The best thing is to use open formats such as RTF (although these also have some problems because of the many versions of it that there are), or XML based formats (OpenOffice generates its own documents in XML), or PDF for read-only documents. We should also highlight recent efforts by the OpenOffice community to create the *standard open document* (used by the suite from versions 2.x), which have made it possible to have a free format as an ISO standard for document creation. This fact has obliged Microsoft to (partially) open its format in versions starting from Office 2007, to incorporate OpenXML formats.

• To provide valid alternatives: the software we stop using has to have alternatives that do the same job as the previous system. Most applications have one or several alternatives with similar, if not better, functionalities. On the Internet you can find different lists of (more or less complete) applications for GNU/Linux that match the functionality of Windows applications.

• Support for running applications for other systems: under some conditions it is possible to run applications for other UNIX systems (with the same architecture, for example, Intel x86), or for MS-DOS or Windows, through compatibility packages or some type of emulator.

Most of the problems that affect desktop migrations are being overcome slowly but surely and will allow us in future to have a larger number of GNU/Linux desktop users, who, as they increase, will have access to better applications encouraging software companies to start implementing versions for GNU/Linux.

In the case of companies, it can be overcome with a gentle migration, starting with servers and workstations, and then desktops after following an extensive training program for users in the new systems and applications.

A process that will help to a large extent is to introduce open code software in education and in public administrations, as in the case of Extremadura region in Spain with its GNU/Linux distribution called Linex; or recent measures for taking this software to primary education, or the measures taken by universities by running courses and subjects using these systems.

5. Migration workshop: case study analysis

In this workshop we will try to apply what we have learned in this unit to analyse some simple migration processes, and some detail of the required techniques (in the case of network techniques, we will look at these in the units on network administration).

We will consider the following case studies:

- Individual migration of a Windows desktop user to a GNU/Linux system.
- Migration of a small organisation with Windows systems and a few UNIX.
- Migration of a standalone Windows server to a Samba server running GNU/ Linux.

5.1. Individual migration of a Windows desktop user to a GNU/Linux system

A user considers migrating to GNU/Linux [Ray02b]. Normally, there will first be a period of cohabitation, so that the user can have both systems and use each one for a series of tasks: tasks will continue to be executed in Windows while the user learns about the new system and finds equivalent software or new software that does tasks for which no software was previously available.

Migration for a private user is perhaps one of the most complex processes; we need to offer users alternatives to what they commonly use, so that adaptation is as simple as possible and the user can adapt gradually and with ease to the new system.

A first possibility would be a dual installation [Ban01] [Sko03b] of the original system (Windows) together with the GNU/Linux system.

Note

Linux Hardware Howto: http://www.tldp.org/HOWTO/HardwareHOWTO/index.html

A first step for a determined machine configuration will consist of checking that our hardware is compatible with Linux [Pri02], either from a list of hardware compatibility or by checking with the manufacturer if new components need to be purchased or the existing ones require a particular configuration. If we are unfamiliar with our hardware, we can check it through the Windows "device administrator" (in the control panel) or using some type of hardware recognition software. At the same time, an advisable method is to use LiveCD-type GNU/Linux distributions, which will allow us to check the functioning of GNU/Linux on our hardware without requiring a physical installation, since the only requirement is the possibility of booting the system from a CD/DVD (in some cases the BIOS configuration may have to be changed for this). There are Live CDs such as Knoppix [Knp] with great support for hardware checks and most GNU/Linux distributions tend to offer a Live CD in order to initially

check its functioning (in some cases, Ubuntu [Ubn] for example, the full installation can be done using the same Live CD). In any case, we should mention that checking with a specific Live CD does not mean that there will not be any problems with the final installation, either because the Live CD is not of the same GNU/Linux distribution that we eventually install or because the versions of the system and/or applications will not be the same.

Regarding the physical installation on disk, we will either need to have unpartitioned free disk space or, if we have FAT/32-type partitions, we can liberate space using programs that make it possible to adjust the size of partitions, reducing an existing partition (a previous data backup here is obviously advisable). Currently, most distributions support various disk partitioning and partition reduction schemes, although problems may arise depending on the distribution. If there is not enough space or there are partitions with file systems that present problems (like NTFS with some distributions), we may have to consider buying a new additional hard disk, to use totally or partially for GNU/Linux.

After checking the hardware, we will have to decide on the distribution of the GNU/Linux system that we will use (a possibility we mentioned before is to choose a Live CD that has been satisfactory and to install that distribution). If the user is inexperienced in GNU/Linux or only has basic computer knowledge, it is preferable to choose one of the more user-friendly distributions such as Fedora, Mandriva, SuSe, or similar (we would highlight the ease of Ubuntu in this regard). If we are more knowledgeable or tempted to experiment, we could try a Debian distribution. In the case of commercial distributions, on most occasions the distributions with compatible hardware (business versions like Red Hat and SuSe certify the hardware that they support), are installed perfectly without any problem and basic configurations are made that allow the operating system to be used immediately. During the process, we will have to install the software, which will normally be defined by sets of oriented software: for servers, specific applications or desktop applications, such as office suites, development applications (if we are interested in programming) etc.

Once the system is installed, we have to tackle the issue of sharing data [Gon00] [Kat01], how will we share the data between the two systems? or is it possible to share certain applications? There are various solutions for this:

- Indirect method: this consists of sharing data using a diskette for example. For this, the best thing are the utilities known as mtools, which allow transparent access to diskettes in MS-DOS format, and there are several commands that function in a very similar way to MS-DOS or Windows. These commands have exactly the same names as the original MS-DOS commands, except that they have an "m" in front, for example: mcd, mcopy, mdir, mdel, mformat, mtype etc.

- Direct method: this consists of using the file system in Windows directly. As we will see in the unit on local administration, GNU/Linux can read and write a large number of file systems, including FAT, FAT32, and NTFS (read only in some cases, although most distributions already include the ntfs-3g [Nt3] driver that allows writing). Mounting the Windows disk is required first and that makes it possible to incorporate the Windows file system into a point of the Linux file tree; for example, we could mount our Windows disk in /mnt/Windows and from this point access its folders and files for reading and writing. With ASCII text files, conversions need to be considered, since UNIX and Windows treat them differently: in UNIX, the end of a line has only one character, the line feed, ASCII 10, whereas Windows has two, the return and the line feed, characters ASCII 13 and 10 (as a curious note, in Mac it is ASCII 13). Which means that usually when we read a DOS/Windows ASCII file, it contains strange characters at the end of a line. There are editors such as emacs that handle them transparently and, in any case, there are GNU/Linux utilities that make it possible to convert them into another format (utilities such as duconv, recode, dos2UNIX, UNIX2dos).

- Use of applications: there are a few alternatives for running the applications (not all of them) for MS-DOS and Windows. For GNU/Linux there are MS-DOS emulators such as Dosemu [Sun02] or DOsBox, and for Windows there is the Wine [Win] software. It can run various Windows applications (for example, it can run some version of Office and Internet Explorer), but it is constantly being improved. If it is vital to run Windows applications, some commercial software can help us; these applications give extra support to Wine, for example, Win4Lin, CrossOver and in some cases special support for games like Cedega. Another potential solution is to use virtual machines; an example of extensively used software is VMware and VirtualBox, which creates a full PC as a virtual machine, simulated by the software, where a large number of different operating systems can be installed. VMware and VirtualBox are available in versions for Windows and for GNU/Linux, which makes it possible to have a GNU/Linux installed with a Windows running on it virtually, or a Windows installed with a virtual GNU/Linux. There are also other solutions of free virtual machines like QEmu, KVM, Bochs. In another segment, virtual machines or generically virtualisation is used oriented at the creation of virtual servers, with solutions such as VMware server or the open projects Xen, OpenVZ, Vserver; where it is possible to make several virtual machines running on an operating system coexist (normally through modifications to the kernel that support this virtualisation), or even directly on the hardware, with small layers of software.
  Aside from sharing the information (applications and/or data) you can search for GNU/Linux applications that replace the original Windows ones as the user gradually learns to use them and sees that they offer the expected functionalities.

Example

A typical case would be the office suite that can be migrated to OpenOffice, which has a high degree of compatibility with Office files and functions fairly similarly, or KOffice (for the KDE desktop), or GNumeric and AbiWord (for Gnome). Or, in the case of image processing, we can take Gimp, with similar functionalities to Photoshop. And numerous multimedia players: Xine, Mplayer (or also a version of RealPlayer). On the Internet we can find numerous lists of equivalent programs between Windows and GNU/Linux.

5.2. Migration of a small organisation with Windows systems and a few UNIX

Migration within an organisation (even a small one) has several difficulties: we will have different work environments and heterogeneous software, and, once more, users who are resistant to change.

Note

For examples of GNU/Linux equivalent applications, see:

http://www.linuxalt.com/
http://wiki.linuxquestions.org/wiki/Linux_software_equivalent_to_Windows_software
http://www.linuxrsp.ru/win-lin-soft/table-eng.htmlç

Now, let's consider an organisation with Windows machines and some UNIX machines as servers or workstations and somewhat "anarchic" users. For example, let's study the following situation: the organisation has a small local network of Windows machines shared by users as equal machines in a Windows workgroup (there are no Windows server domains).

The group is diverse: we have machines with Windows 98, ME, NT, XP, but configured for each user with the software needed for their daily jobs: whether Office, a browser, e-mail reader, or development environments for different language programmers (for example, C, C++, Java).

There are some extra hardware resources available, such as various printers connected to the local network (they accept TCP/IP jobs), which can be used from any point within the organisation. At the same time, there is a shared machine, with a few special resources, such as a scanner, CD recorder and directories shared by the network, where users can leave their own directories with their files for backup processes or to recover scanned images, for example.

We also have several workstations, in this case Sun Microsystem's SPARC, which are running Solaris (commercial UNIX of Sun). These stations are dedicated to development and to some scientific and graphics applications. These machines have NFS services for file sharing and NIS+ for handling the information of users who connect to them and who can do so from any machine in a transparent manner. Some of the machines include specific services; one is the company's web server and another is used as an e-mail server.

We are considering the possibility of migrating to GNU/Linux because of an interest in software development and the particular interest from some users to use this system.

Also, the migration will be made the most of in order to resolve certain problems related to security – some old Windows systems are not the best way of sharing files; we want to restrict use of the printer (the cost in paper and associated costs are high) to more reasonable quotas. At the same time we would

like users to have a certain amount of freedom, they will not be obliged to change system, although the suggestion will be made to them. And we will also take advantage in order to purchase new hardware to complement existing hardware, for example if the workstations require additional disk space, which imposes limits on e-mail and user accounts.

Following this small description of our organisation (in other more complex cases it could fill several pages or be a full document analysing the present situation and making future proposals), we can start to consider the possibilities for solving all this:

- What do we do with the current workstations? The cost of maintenance and software licenses is high. We need to cover the maintenance of faults in the stations, expensive hardware (in this case, SCSI disks) and also expensive memory extensions. The cost of the operating system and its updates is also expensive. In this case, we have two possibilities (depending on the budget that we have to make the change):
  - We can cut costs by converting the machines to GNU/Linux systems. These systems have a SPARC architecture and there are distributions that support this architecture. We could replace the services for their GNU/Linux equivalents; replacement would be virtually direct, since we already use a UNIX system.

  - Another possibility would be to eliminate Sun's proprietary hardware and to convert the stations into powerful PCs with GNU/Linux; this would make subsequent maintenance simpler, although the initial cost would be high.

- And what about the workstations software? If the applications have been developed in-house, it may be enough to compile them again or to make a simple adjustment to the new environment. If they are commercial, we will have to see whether the company can provide them in GNU/Linux environments, or if we can find replacements with a similar functionality. In the case of the developers, their environments of C, C++ and Java languages are easily portable; in the case of C and C++, gcc, the GNU compiler, can be used and there are numerous IDEs for development (KDevelop, Anjuta,...); or in the case of Java, the Sun kit can be used in GNU/Linux and in various open code environments (IBM's Eclipse or Netbeans).

- And what about users? For those who are interested in GNU/Linux systems, we can install dual equipment with Windows and GNU/Linux so that they can start to test the system and if they are interested, we can finally transfer to just the one GNU/Linux system. We can find two types of users: purely office suite users, who will basically need the suite, navigator and e-mail; all of which can be offered with a GNU/Linux desktop such as Gnome or KDE and software such as OpenOffice, Mozilla/Firefox navigator, and Mozilla Mail or Thunderbird e-mail (or any other Kmail,

Evolution...). They are more or less directly equivalent, it all depends on users' desire to test and use the new software. For developers, the change can be more direct, since they are offered many more environments and flexible tools; they could pass completely over to the GNU/Linux systems or work directly with the workstations.

- And the printers? We could establish a workstation as a printer server (whether through TCP/IP queues or Samba server), and control printing by means of quotas.

- The shared machine? The shared hardware can be left on the same machine or can be controlled from a GNU/Linux system. Regarding the shared disk space, it can be moved to a Samba server that will replace the current one.

- Do we expand the disk space? This will depend on our budget. We can improve control by means of a quota system that distributes space equitably and imposes limits on saturation.

5.3. Migration of a standalone Windows server to a Samba server running GNU/Linux

The basic required process tends to me much more extensive, consult the bibliography for the full steps to be taken.

In this case, the basic required process for a migration from a Windows server that shares files and a printer to a Samba server in a GNU/Linux system.

Thanks to software such as Samba, migration from Windows environments is very flexible and fast and even improves the machine's performance.

Let's suppose a machine belonging to a workgroup GROUP, sharing a printer called PRINTER and with a shared file called DATA, which is no more than the machine's D drive. Several Windows clients access the folder for reading/writing, within a local network with IP 192.168.1.x addresses, where x will be 1 for our Windows server, and the clients will have other values (192.168.x.x networks are often used as addresses to install private internal networks).

As part of our process we will build a Samba server, which is what, as we saw, will allow us to run the SMB/CIFS (server message block / common Internet file system) protocol in GNU/Linux. This protocol allows the file system and the printers to interact through networks on different operating systems. We can mount folders belonging to Windows on the GNU/Linux machines, or

part of the GNU/Linux folders on Windows and similarly with each other's printers. The server consists of two daemons (system processes) called smbd and nmbd.

The smbd process manages clients' requests from shared files or printers. The nmbd process manages the machines' names system and resources under the NetBIOS protocol (created by IBM). This protocol is independent from the network used (currently, in NT/2000/XP Microsoft generally uses Netbios over TCP/IP). The nmbd also offers WINS services, which is the name assignment service that is normally run on Windows NT/Server if we have a collection of machines; it is a sort of combination of DNS and DHCP for Windows environments. The process is somewhat complex, but to summarise: when a Windows machine starts up or has a static IP address or dynamic address through a DHCP server and additionally possibly a NetBIOS name (that the user assigns to the machine: in network identification), then the WINS client contacts the server to report its IP; if a network machine subsequently requests the NetBios name, the WINS server is contacted to obtain its IP address and communication is established. The nmbd runs this process on GNU/Linux.

Like any other network service, it should not be run without considering the risk activating it could entail, and how we can minimise this risk. Regarding Samba, we need to be aware of security issues, because we are opening part of our local or network files and printers. We will also have to check the communication restrictions properly in order to prevent access to unwanted users or machines. In this basic example, we will not comment on these issues; in a real case scenario, we would have to examine the security options and only allow access for those we want.

In the migration process, we will first have to configure the GNU/Linux system to support Samba [Woo00], we will need the Samba file systems support in the kernel (*smbfs*), which is normally already activated. We should add that currently there is additional support in the kernel through the cifs module [Ste07], which as of kernel version 2.6.20 is considered the default method, leaving smbfs as a secondary option. The cifs module offers support for new features related to the CIFS protocol (as an extension of SMB). Through "smbfs" and "cifs" file system names these modules allow us to conduct operations for mounting Windows file systems onto the Windows directory tree (mount -t smbfs or mount -t cifs). Apart from the fact that the kernel support is inclined towards the cifs module, there are some characteristics that may need smbfs support, which means that usually both modules are activated in the kernel. We should also mention the configuration issue, whereas smbfs bases its functioning on the Samba configuration (as we will see in the smb.conf file), the cifs module is given its configuration through the operations (for example, in the mounting process through mount).

In the case of using a Samba server, in addition to the kernel support, we will need to install the associated software packages: we will have to examine what packages related to Samba the distribution includes and install those associated to the functioning of the server. And also, if wanted, those related to Samba as a client, in the event we wish to be clients of Windows machines or to test resources shared with the Windows machines from our GNU/Linux system. In a Debian distribution, these packages are: samba, samba-common, smbclient, smbfs. It may also be interesting to install swat, which is a web-based graphics tool for Samba services administration. For our GNU/Linux Samba server [Woo00] [War03], for the proposed example, we will have to transfer the contents of the previous D disk (where we had our shared file system) from the original machine to the new machine and place its content in a path, like, /home/DATA, whether through a backup copy, FTP transfer, or using Samba as a client to transfer the files.

Regarding the use of GNU/Linux as a Samba client, it is fairly simple. Through the use of client commands for occasional use of the file system:

a) We mount a Windows shared directory (for instance, host being the name of the Windows server), on an existing predefined mounting point:

```
smbmount //host/carpeta /mnt/windows
```

b) We will place the access to the Windows folder of the host machine in our local directory, accessing in the directory tree:

```
/mnt/windows
```

c) Next, when it is no longer in use we can dismount the resource with:

```
smbumount /mnt/windows
```

If we are not aware of the shared resources, we can obtain a list with:

```
smbclient -L host
```

And we can also use smbclient //host/folder, which is a similar program to an FTP client.

In the event of wanting to make the file systems available permanently, or to provide certain special configurations, we can study the use of *mount* directly (the smbxxxx utilities use it), whether with the smbfs or cifs file systems (supported in the kernel), taking the parameters into account (Windows users/groups authentication or other service parameters) that we will have to provide depending on the case, and of the pre-existing Samba configuration [Ste07].

Note

Always consult the **man** pages, or manuals, that come with the software package.

In the case of the Samba server, once we have installed all the Samba software, we will have to configure the server through its configuration file. Depending on the version (or distribution), this file may be in /etc/smb.conf or in /etc/samba/smb.conf. The options shown here belong to a Samba 3.x.x installed on a Debian distribution system. Other versions may have a few minor modifications.

During the installation of the software packages we will normally be asked for data regarding its configuration. In the case of Samba, we will be asked for the workgroup to be served; we will have to place the same group name as in Windows. We will also be asked if we want encrypted passwords (advisable for security reasons, in Windows 9x they were sent in raw text, in a clear case of scarce security and high system vulnerability).

Next we will look at the process of configuring the file smb.conf. This file has three main sections:

1) *Global* (basic functioning characteristics).

2) *Browser* (controls what other machines see of our resources).

3) *Share* (controls what we share).

In this file's extensive manual we can see the available options (man smb.conf). We will edit the file with an editor and see some of the file's lines (characters '#' or ';' at the beginning of a line are comments: If the line contains ';' it is a comment; to enable a line, if it is an optional configuration line we must edit it and remove the ';'):

```
workgroup = GROUP
```

This shows the Windows workgroup that the Windows client machines will be members of.

```
server string = %h server (Samba %v)
```

We can place a text description of our server. The *h* and the *v* that appear are variables of Samba that refer to the host name and version of Samba. For security reasons, it is a good idea to remove the *v,* since this will inform the exterior what version of Samba we have; if there are known security bugs, this can be used.

```
hosts allow = 192.168.1
```

This line may or may not be present, and we can include it to enable what hosts will be served; in this case, all of those in the 192.168.1.x range.

```
printcap name = /etc/printcap
```

The printcap file is where GNU/Linux stores the printers' definition, and this is where Samba will look for information about them.

```
guest account = nobody
```

This is the guest account. We can create a different account, or just enable access to Samba for the users registered on the GNU/Linux system.

```
log file = /var/log/samba/log.%m
```

This line tells us where the Samba log files will be stored. One is stored per client (variable *m* is the name of the connected client).

```
encrypt passwords = true
```

For security reasons it is advisable to use encrypted passwords if we have client machines with Windows 98, NT or above. These passwords are saved in a /etc/samba/smbpasswd file, which is normally generated for users of the Samba installation. Passwords can be changed with the *smbpasswd* command. There is also an option called *UNIX password sync*, which allows the change to be simultaneous for both passwords (Samba user and Linux user).

Next, we will jump to the Share Definitions section:

```
[homes]
```

These lines allow access to the users' accounts from the Windows machines. If we don't want this, we will add some ';' to the start of these lines, and when the machines connect they will see the name comment. In principle, writing is disabled, to enable it, you just have to set "yes" as the writable option.

Any sharing of a specific directory (Samba tends to call a group of shared data a partition), we will proceed as shown in the examples that appear (see, for example the definition of sharing the CD-ROM in the lines that start with [cdrom]). In path we will place the access route.

Example

In our case, for example, we would give the name DATA to the partition on the route /home/DATA, where we had copied the D disk from the original Windows machine and the path where it can be found, in addition to a large group of options that can be modified, users authorised to access them and the way of doing so.

Note

See: man smb.conf

There is also a profiles definition, that makes it possible to control the profiles of Windows users, in other words, the directory where their Windows desktop configuration is saved, the start up menu etc.

The method is similar for the printers: a partition is made with the printer name (the same one given in GNU/Linux), and in the path we place the queue address associated to the printer (in GNU/Linux we will find it in: /var/spool/samba/PRINTER). And the option printable = yes, if we want jobs to be sent with Samba. And we can also restrict user access (valid users).

Once we have made these changes we will just have to save them and reinitiate Samba so that it can read the new configuration. In Debian:

```
/etc/init.d/samba restart
```

Now, our shared directory and the printer through Samba will be available to serve users without them noticing any difference in relation to the previous connections with the Windows server.

Activities

1) In the GNU/Linux services description, do we find we are missing any functionality? What other type of services would we add?

2) In the second case study of the tutorial (the one of the organisation), how would you change the IT infrastructure if you had zero budget, an average budget, or a high budget? Present some alternative solutions to the ones shown.

3) Virtualisation technologies like VMware Workstation or VirtualBox, virtual machine through software, which can install operating systems on a virtual PC. You can obtain the software from www.vmware.com or www.virtualbox.org. Test (in the case of having a Windows license) installing it on Windows, and then on GNU/Linux on the virtual PC (or the other way around). What advantages does this method for sharing operating systems offer? What problems does it cause?

4) If we have two machines for installing a Samba server, we can test the server installation or configuration in configurations of Samba UNIX client-Windows server, or Windows client-Samba server in GNU/Linux. You can test it on a single machine using the same machine as a Samba server and client.

Bibliography

Other sources of reference and information

[LPD] Linux Documentation Project offers Howtos regarding different aspects of a GNU/Linux system and a set of more detailed manuals.

[Mor03] Good reference for the configuration of Linux systems, with some case studies in different environments; comments on different distributions of Debian and Red Hat.