

Seminario de Introducción a Linux

RAXi - RAX Informática
Radioclub Alfonso X El Sabio¹

16 de mayo de 2000

¹Universidad Alfonso X El Sabio – Villanueva de la Cañada

(c) Copyright 2000 RAX, Radioclub Alfonso X el Sabio, <rax@uax.es> Sección RAXi, RAX Informática.

Juan Jesús Acevedo <jacevedo@arrakis.es>
Antonio Estrada <azayases@alum.uax.es>
Ignacio Grande <igranola@alum.uax.es>
Daniel Payno <payno@gul.uc3m.es>
Jaime Robles <ea4abw@amsat.org>

El siguiente documento está suscrito a la licencia abierta de Openresources para escritos y similares^{1 2}

Inicio de la licencia

Este documento puede ser libremente leído, almacenado, reproducido, distribuido, traducido o citado de cualquier forma y en cualquier soporte siempre que se cumplan las siguientes condiciones:

1. Todo lector o usuario de éste documento reconoce que es consciente de que no se le ofrece ninguna garantía en cuanto a sus contenidos, en ningún respecto, y particularmente en lo concerniente a su veracidad, precisión y adecuación para cualquier propósito.
2. No se le hará ninguna otra modificación más allá de la estética, cambio del formato de presentación, traducción, corrección de errores sintácticos obvios, o los permitidos por las cláusulas subsiguientes.
3. Se pueden insertar comentarios u otros añadidos siempre que aparezcan claramente como tales; las posibles traducciones o fragmentos han de referirse claramente a una versión original completa, preferentemente una que sea de fácil acceso cuando sea necesario.
4. Las posibles traducciones, comentarios y demás añadidos o modificaciones han de estar fechados e identificado(s) su autor(es) (posiblemente con un alias).
5. Esta licencia está protegida y se aplica a la totalidad del documento incluyendo sus modificaciones y adiciones (excepto para breves citas), independientemente del formato de presentación.
6. Cualquier referencia a la “versión oficial”, “versión original”, o “cómo obtener las versiones originales” del documento esta protegida palabra por palabra. Cualquier aviso de copyright del documento estará protegido palabra por palabra. Además, el título y el(los) autor(es) del documento original han de ser claramente mencionados como tales.
7. En el caso de traducciones, las frases íntegras mencionadas en el (6.) serán conservadas en el lenguaje del documento original acompañadas por traducciones íntegras al lenguaje del documento traducido. Todas las traducciones indican claramente que el autor original no es responsable del texto traducido. Esta licencia será incluida al menos en el lenguaje en el cuál está referida en la versión original.
8. Cualquiera que sea el medio de almacenamiento, reproducción o distribución, cualquier persona con acceso a una versión digitalizada de éste documento ha de ser capaz de hacer una copia digitalizada en un formato directamente utilizable, y si es posible, editable, de acuerdo con los estándares aceptados y publicamente documentados.
9. La redistribución de este documento a una tercera persona requiere redistribución simultánea de ésta licencia, sin modificación alguna, y en particular sin ninguna otra condición o restricción, expresa o derivada, relacionada o no con ésta redistribución. En particular, en el caso de inclusión en una base de datos o colección, el dueño o el administrador de la base de datos o la colección renuncia a cualquier derecho relacionado con esta inclusión y que concierna a los posibles usos del documento tras su extracción de dicha base de datos o colección, tanto de forma individual como acompañada de otros documentos.

Cualquier incompatibilidad con las cláusulas precedentes con decisiones o limitaciones legales, contractuales o judiciales implica una correspondiente limitación de los derechos de lectura, uso, o distribución de éste documento, ya sea íntegro o modificado.

Fin de la Licencia

¹<http://www.openresources.com/magazine/license>

²Traducción de Daniel Payno, <payno@gul.uc3m.es>

Índice General

1	Introducción a Linux	1
1.1	¿Qué es Linux?	1
1.1.1	Sistemas Operativos	1
1.2	Génesis de Linux	1
1.2.1	GNU. “GNU’s Not Unix”	1
1.2.2	Internet y el desarrollo de Linux	2
1.3	Distribuciones de Linux	2
1.3.1	Red Hat	2
1.3.2	Debian	2
1.3.3	S.u.S.E.	3
1.3.4	RedHat Eurielec	3
1.3.5	Otras	3
1.4	Variantes	3
1.5	Características de Linux como Sistema Operativo	3
1.6	Últimas Tecnologías	5
1.6.1	IrDA	5
1.6.2	DVD	5
1.6.3	USB	5
1.6.4	UDMA	5
1.6.5	IPv6	5
2	Comandos Básicos	7
2.1	Ayuda en pantalla	7
2.2	El sistema de Archivos de Linux	7
2.2.1	Permisos de los Archivos	7
2.2.2	Dispositivos	8
2.2.3	El Árbol de Directorio	9
2.3	Intérprete de Comandos (BASH - Bourne Again Shell)	10
2.3.1	Archivos de configuración	10
2.4	Comandos más usuales	11
3	El Sistema XWindow	13
3.1	¿Qué son las Xwindow?	13
3.2	Instalando las X	13
3.2.1	Consiguiendo las X	13
3.2.2	Ahora a instalar...	14
3.2.3	“Castellanizando” las X	14
3.2.4	Apariencia	17
3.2.5	Arrancando las X	17
3.3	Modelo Cliente/Servidor	17
3.3.1	Introducción	17
3.3.2	Servidor de X (XFree86)	18
3.3.3	Clientes de X	18
3.3.4	Gestores de ventanas (window managers-WM)	18

Apéndice B

Direcciones de Interés

Existen multitud de organizaciones, grupos, personas, sitios, etc ... a los que acudir para seguir informándonos en cuanto a Linux.

RAX <http://rax.uax.es>

GUL, Grupo de Usuarios de Linux de la Carlos III <http://www.gul.uc3m.es>

FTP del GUL <ftp://ftp.gul.uc3m.es>

Jaime <http://www.geocities.com/SiliconValley/5161/linux.html>

LUCas <http://lucas.hispalinux.es>

Linux Headquarters <http://www.linuxhq.com>

Kernel <http://www.kernel.org>

Linux <http://www.linux.org>

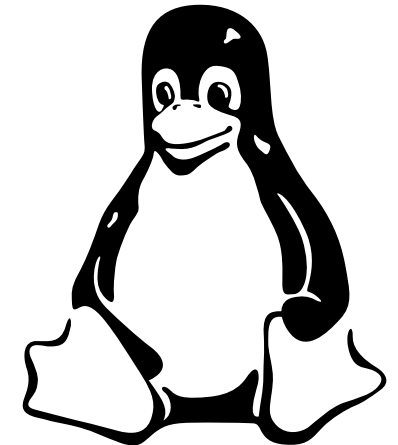
Freshmeat <http://www.freshmeat.com>

Linux Today <http://linux.today.com>

Debian <http://www.debian.org>

RedHat <http://www.redhat.com>

...



... Y este es Tux, la mascota de Linux ...

3.4	Aplicaciones para las X	19
3.4.1	Office	20
3.4.2	Emuladores	20
3.4.3	Gráficos	21
3.4.4	Utilidades	22
3.4.5	Redes	23
3.5	Para los radioaficionados	24
3.6	Trucos	24
3.7	¿Necesitas algo más.? Prográmalo tu mismo!!! :-P	25
4	Introducción a las redes TCP/IP	27
4.1	Enredando	27
4.1.1	Internet	27
4.2	Unix y las redes de ordenadores	27
4.2.1	Linux en red	28
4.2.2	Interfaces de red	28
4.2.3	Modelo cliente-servidor	29
4.2.4	Direcciones de Internet	29
4.2.5	Protocolos	29
4.2.6	DNS	29
5	Configurando el interfaz de red	31
5.1	Introducción	31
5.2	PPP, Una ventana abierta a la Red	31
5.2.1	Configuración del puerto serie	32
5.2.2	Configuración del demonio pppd	33
5.3	PLIP, una red para pobres	34
5.3.1	Conseguir el cable	35
5.3.2	Instalación de PLIP	35
5.3.3	Un ejemplo real	35
5.4	Tarjetas de red (Ethernet)	37
5.4.1	Instalar la tarjeta	37
5.4.2	Configurar la red	38
5.4.3	Notas	38
5.5	El protocolo AX25	39
6	Trabajando en red	41
6.1	Comandos útiles	41
6.1.1	Ping	41
6.1.2	Traceroute	42
6.1.3	Finger	42
6.1.4	Netstat	43
6.2	Telnet	44
6.3	FTP	44
6.4	Servidores de red	45
6.4.1	El fichero /etc/services	45
6.4.2	Puertos reservados	45
6.4.3	El programa /etc/inetd	45
7	Entornos de Desarrollo	47
7.1	Lenguajes Soportados	47
7.2	Programación en C en linux	47
7.2.1	El gcc	48
7.2.2	El gdb	48
7.2.3	Ejemplo. Programación en C	48
7.2.4	Make	49
7.2.5	Formato de los ejecutables	49

7.3	Programación en Java en Linux	49
7.3.1	Ejemplo. Programación en Java	50
7.4	GTK y el GLADE	50
7.5	Compilar paquetes bajados de Internet	50
7.6	IDE, Entornos de Desarrollo Integrados	51
8	Seguridad del sistema	53
8.1	Introducción a la seguridad	53
8.2	Planificando las necesidades de seguridad	53
8.2.1	Conocimiento de los riesgos	53
8.3	Claves de usuario	54
8.4	Secure Shell (ssh)	55
8.4.1	Instalación	55
8.4.2	Uso del programa	56
8.5	Los registros del sistema (logs)	57
8.6	Eliminar servidores innecesarios	57
8.7	Posibles amenazas informáticas	57
8.8	Últimos consejos	58
A	Ejemplo de Instalación	I
A.1	¿Por dónde empiezo?	I
A.1.1	¿Qué es una partición? ¿Cómo hago yo una?	I
A.2	Tipos de Instalación	II
A.2.1	Típica:	II
A.2.2	Comoda:	II
A.3	Instalación paso a paso	II
B	Direcciones de Interés	V

5. 5ª pantalla Inicializar una partición swap

Ahora hemos de indicarle cual es la partición que queremos que use como swap y que nos la formatee. Para ello ojearemos el papel que hemos copiado con la tabla de partición al crear las particiones, fijándonos en el tamaño, pues aún no sabemos el criterio de nombre para las particiones en Linux., si bien, al haberle indicado el tipo en el `fdisk` al crearla, el programa de instalación nos la reconoce como la partición de swap.

6. 6ª pantalla Inicializar una partición para el raíz/ Procedemos igual que en el caso anterior, y el igual que en él, nos formatea la partición, preguntándonos si queremos una comprobación exhaustiva para detectar errores. Le decimos que sí, y elegimos la partición que hemos creado para que lo haga.

7. 7ª pantalla Montar el arbol raíz/ Una vez inicializada la partición donde va a ir el raíz, lo montamos.

8. 8ª pantalla Instalar el Sistema Operativo/ En esta pantalla pasamos a una serie de subpantallas, que no pediran de dónde instalarlo.

localización en nuestro caso, lo instalamos desde un `cdrom`

localización del CD en nuestro caso, estamos usando el CD que está como esclavo en el primer canal IDE primario, luego el nombre es el `/dev/hdb`

Directorio Base de la instalación Directorio en el CD donde se encuentra la instalación, en nuestro caso `/debian`

Lista de paquetes Le decimos que la busque, y sin problemas, aceptar

Configuración de Módulos Ahora nos va a pedir configurar una serie de módulos específicos nuestros para que los añada al kernel que va a instalar en nuestro sistema y nos funcione todo el HW perfectamente de primeras. Vamos buceando y activando los módulos que creamos necesarios.

En nuestro caso, el de la impresora, nuestro mouse (ps2), los puertos serie, los códigos de página (cp 850 y iso-8859-1), y quizás el de nuestra tarjeta de red, además de los protocolos de Red, como el `ppp` e `ipv4`

Configuración de la Red Nos pregunta si estamos conectados a una red. Como no es habitual, respondemos no.

9. 9ª pantalla Instalación de Sistema Operativo Base/ De igual forma, hemos de especificar de dónde ha de instalarlo, es decir, `cdrom`, `/dev/hdb`, lista . . .

10. 10ª pantalla Configuración del sistema Base/ Pues eso.

11. 11ª pantalla Boot, arranque/ Aquí hemos de elegir de crear la entrada de LILO, para lo cual tenemos dos opciones, el MBR del disco, o el sector 0 de la partición de Linux. La opción más segura es la 2ª, aunque, personalmente, la primera a mí nunca me ha dado problema alguno. Luego crearemos un disquette de arranque, por si acaso. Una vez hecho esto, reorganizaremos la máquina.

Más tarde haremos `'man lilo.conf'` para poder configurar el LILO para el arranque de la partición de Windows que hemos dejado en el disco.

12. 12ª Reinicio Instalación de Paquetes/ Una vez reiniciados, nos dará la bienvenida y nos pedirá cambiar la contraseña de root. A la vez, nos instará a crear un usuario normal, pues root sólo ha de usarse cuando sea imprescindible.

La siguiente pregunta es que se ha dado cuenta de que no tenemos PCMCIA, y nos sugiere quitarlo del kernel. Bien.

13. PPP-Config/ Y ahora, como hemos activado el módulo de PPP, nos manda directamente a una utilidad para configurar nuestra conexión PPP con nuestro servidor de Internet (ISP), pidiéndonos el tlf, la posición del módem, el nombre de usuario, nuestra contraseña, los DNS, etc..

14. Instalación de paquetes - Paquetes reconfigurados/ Ahora, ejecutamos `dselect` como nos dicen, y nos preguntan si queremos una instalación prefijada entre unas cuantas. Le decimos que sí, y elegimos la de *Scientific Workstation (~500M)*, seguimos, los pasos, y nos quedamos delante de la pantalla embobados por si nos pregunta cualquier cosa de configuración de algun paquete , que lo hará (eso sí, con nuestro café en la mano). Si todo va bien, en menos de una horita habrá terminado, y no tendremos que reiniciar la maquina para tener todo configurado.

Otra cosa que conviene hacer es entrar en la BIOS y desactivar el soporte de chequeo de Virus, pues este creará que el LILO lo es, y no nos dejará arrancar.

Haciendo hueco

Podemos redimensionar la partición que tenemos con Windows sin borrarla, y luego crear las dos que necesitamos en el espacio de disco que hemos liberado. Para ello podemos usar un programa llamado FIPS que traen todos los CD's de distribuciones de Linux que yo conozco, o herramientas ya comerciales, y más amigables (pero ya sabéis, acouinando) como el Partition Magic 4.0, que incluso permite redimensionar particiones de Linux sin pérdida de datos. También existe una herramienta GNU llamada parted que permite hacer eso, pero sigue en desarrollo.

En cualquiera de los dos casos, Backup obligado antes de nada. Las instrucciones y uso del FIPS son muy claras y sencillas, luego huelga que las repitamos aquí.

A.2 Tipos de Instalación

Bien, ya tenemos el espacio libre hecho con el FIPS, donde vamos a meter las dos particiones, la de Linux, y la destinada a SWAP. Además, no le hemos dado formato en DOS. Tampoco pasa nada si las hemos formateado, pero es más sencillo a la hora de la instalación.

Ahora empezamos con la instalación, para lo cual también hay dos formas:

A.2.1 Típica:

Mediante la creación de dos disquettes, uno de arranque, y otro con la instalación, llamados BOOT y RAM disk (o image disk), respectivamente. Todo esto viene, como el uso del FIPS; bien documentado en el CD de la distribución.

A.2.2 Comoda:

Con un *install.bat* o *boot.bat* que suele venir en el CD, si bien generalmente, bastante escondido.!

Nosotros lo vamos a hacer a la manera cómoda, para ahorrarnos disquettes. Y por que así nos dará menos problemas la detección del CD-ROM, pues ya tendremos el driver de MS-DOS para el CD cargado en memoria. Para ello hemos de estar en MS-DOS, con el controlador del CD cargado, y bucear por el CD hasta el directorio donde este el .bat.

A.3 Instalación paso a paso

1. 1ª pantalla

Lo primero que nos pregunta es si queremos la instalación (gráfica) a color o en blanco y negro. Elegimos color y seguimos.

2. 2ª pantalla

Nos aparece un texto explicativo de la Debian que estamos instalando. LEERLO

3. 3ª pantalla Configurar el teclado

Entramos en la selección del teclado y elegimos 'es'/'es-qwerty' con la barra espaciadora, que es el mapa del teclado correspondiente al nuestro.

4. 4ª pantalla Particionar el disco duro

Con el FIPS hemos creado espacio libre, y ahora tenemos que particionarlo. Básicamente, entraremos en el cfdisk (programa para particiones de Linux) que es como el fdisk de DOS pero MUCHO más potente. Miramos el espacio libre que hemos creado y creamos una partición nueva asignándole de 64-128 Mb de espacio. Posteriormente, le cambiamos el tipo para que sea del 82, ó Linux Swap.

Hacemos lo mismo que el espacio libre sobrante, pero sin tener que cambiarle el tipo, pues el programa nos lo asigna a Linux (83) directamente.

Capítulo 1

Introducción a Linux

Hola a todos y bienvenidos a este curso organizado por la sección del RadioClub Alfonso X El Sabio dedicada a la informática y la telemática, aplicadas a la radioafición; el RAXi. Como parte del RAX os invitamos a todos a que no os quedéis aquí, en Linux, y os acerquéis a cualquier reunión del RAX, ó a nuestro local en el último piso del edificio C, a enteraros del tema.

1.1 ¿Qué es Linux?

Probablemente la gran mayoría de los que venís aquí haya oído la palabra linux a lo largo del último año. Y eso por que Linux, este pasado año, ha empezado a salir de su sombra de Sistema Operativo de locos informáticos y gurus con tiempo libre, para llegar al entorno doméstico (aunque poquito a poco, :), e incluso saltar a las noticias en periódicos de prestigio. Seguro, que los más se la hayan oído al típico compañer@, colega, o novi@, que de repente se ha puesto a hablar maravillas de no sé que 'inus'... X'D Pues bien, en este cursillo vamos a adentrarnos en Linux. Primero, claro está, os diremos ¿qué es Linux?, los fundamentos (donde hablaremos de cosas muy relacionadas con Linux pero distintas como es el GNU), y demás zarandajas, para que la próxima vez que oigáis hablar de Linux seáis vosotros los que hablan, XXX"DDDD.

1.1.1 Sistemas Operativos

Ni pretendemos analizar el Sistema Operativo Linux, ni definirlo con todo su rigor. Sin embargo hay que decir qué es lo que es Linux. Es un sistema operativo. Windows 95, Windows 98, DOS y Windows NT, además de Unix, son también sistemas operativos. Es lo necesario para que el ordenador funcione. Aquello que 'sabe hablar con el ordenador', a nivel físico, de resistencias y demás... Además, Linux es un sistema operativo compatible con la norma más generalizada del sistema operativo Unix (la norma POSIX).

1.2 Génesis de Linux

Linux se creo como proyecto de un estudiante finlandés , Linus Torvalds, que usó para ello un sistema operativo de uso didáctico creado por Andrew S. Tannenbaun, minix, inspirado en el Unix. Lo que entonces era Linux es lo que actualmente se conoce como el kernel. El resto de los programas, que conforman en su conjunto Linux, provienen de otras fuentes, como son la GNU y la Free Software Foundation.

1.2.1 GNU. "GNU's Not Unix"

Richard Stallman, el gran profeta del software libre, desde su FSF¹, creó este movimiento con la ambición de implementar un sistema operativo Unix con software 100% libre, siguiendo la patente creada por ellos para las fuentes del software, la GPL. La GPL, ó General Public License, en pocas palabras, es una licencia que exige que los programas sujetos a ella se distribuyan conjuntamente con su código fuente, además de obligar a que cualquiera que modifique las fuentes (para la mejora del programa) distribuya el programa mejorado incluyendo sus modificaciones. Sin embargo, cuando

¹Free Software Foundation

apareció el kernel Linux, la 'madre' del sistema operativo, el HURD, nombre clave del kernel para el GNU, no estaba (ni lo sigue estando, aunque ya es lo suficientemente estable para que se vaya a incluir en lo que será la nueva versión de la distribución Debian, la 2.2) terminado. Hará un par de años, la FSF adoptó el kernel linux como el kernel para su proyecto GNU de forma oficial (aunque a regañadientes), pero sin abandonar el desarrollo del HURD. Es por eso que hoy en día Linux y GNU son términos que van de la mano, denominándose los sistemas como GNU“Linux.

1.2.2 Internet y el desarrollo de Linux

Es Linux el mejor ejemplo actual de la potencia que posee internet a la hora del trabajo en grupo por gente de distintas partes del mundo pues en gran medida su increíble desarrollo se debió al gran éxito que tuvieron las primeras versiones de Linux que Linus publicó en la red. El foro de donde todo partió fue el grupo de noticias de Internet COMP.OS.MINIX. Todos los hackers² del área se entusiasmaron con el proyecto, y, gracias a que Linus lo suscribió a la forma de la GPL, es decir lo convirtió en un sistema operativo libre, sin necesidad de tener que pagarle dinero a una empresa desarrolladora para obtener de ella un licencia de uso, se pusieron manos a la obra en mejorarlo.

No se tardó mucho en obtener una versión 1.0, pero la gran revolución vino con la aparición de la versión 2.0 del kernel. En Linux, y por extensión en muchos de los proyectos de programación para el mismo, se eligió una convenio para distinguir las versiones estables del kernel de las inestables, dado que las versiones aparecían con un intervalo de tiempo muy pequeño. Las versiones estables se distinguen por un número par de la versión en el segundo número de la misma, como 2.0.xx o 2.2.xx, y las inestables, por uno impar como 2.1.xx, siendo xx la versión actual dentro de cada campo. A día de hoy, la última versión estable del kernel es la 2.2.14, y de la inestable mejor no decirlo, por que seguro que en el camino a la imprenta de éste documento, ya habrán salido 2 ó 3. La versión 2.2.x del kernel apareció a finales de febrero del 1999, introduciendo mejoras importantes.

1.3 Distribuciones de Linux

Como tal, Linux no es un único Sistema Operativo. Lo único fijo, es el kernel. El resto de las aplicaciones, variantes, programas de correo, de edición (como el StarOffice 5.0, el Wordperfect 8, klyx/lyx, o L^AT_EX, con el que está escrito este documento), forman parte de la grandísima cantidad de programas disponibles para Linux, incluyendo de empresas comerciales, como Caldera, Oracle, Stardivision, etc.. que ya tienen versiones linux de sus programas más importantes o están en proceso de sacarlas al mercado, con una cláusula que las hace gratuitas para uso personal (similar a la licencia adoptada por Netscape). Si bien el ámbito de donde provienen la mayoría de las aplicaciones distribuidas para Linux es el proyecto GNU, ya explicado, por lo que suele llamarsele con propiedad el Sistema Operativo GNU/Linux. Es por ello que existen distintos grupos de gente, ahora ya empresas con unos ingresos algo sustanciales, que se dedican a compilar los programas que ellos consideran más interesantes, y ofrecerlo conjuntamente con Linux en lo que se conoce como una distribución de Linux. Cabe destacar un par de ellas.

1.3.1 Red Hat

Red Hat es una empresa americana que ha ayudado en gran medida a que el entorno profesional aprecie las virtudes de Linux, ofreciendo apoyo y asistencia técnica a las empresas que así lo deseen. Su jompeich es: <http://www.redhat.com>, y su última versión disponible es la 6.1.

1.3.2 Debian

Debian no es una empresa, por lo menos al uso, pues se compone de multitud de internautas dispersos por el mundo que participan a la hora de desarrollar su distribución. Cabe destacar que se caracterizan por incluir exclusivamente software para Linux que esté sujeto a la GPL. Su última versión disponible es la 2.1, <http://www.debian.org>

²Los periodistas en general, confunden y crean confusión con el término 'hacker'. Hacker es un término que se acuñó en las Universidades americanas en la década de los 60. Este hacía referencia a los primeros informáticos, que, de tanta pasión que ponían en su afición, no dudaban en modificar el esquema eléctrico de su ordenador, las fuentes, crear programas, etc.. con tal de mejorar el funcionamiento del mismo. Hoy en día se conoce por hacker a aquél que es un apasionado de la informática y busca un conocimiento a fondo, desde varios frentes, en la misma. Aquellos delincuentes informáticos, hackers al fin y al cabo, que utilizan estos conocimientos para cometer delitos de privacidad, intrusión e incluso robo en redes telemáticas ajenas (con fines delictivos o deportivos -con afán de superación-) son los denominados 'crackers'. Y no conviene confundirlos.

Apéndice A

Ejemplo de Instalación

A continuación voy a relatar los pasos que se dan para instalar la versión 2.0 de la distribución GNU/Linux Debian. La elección de ésta distribución, a priori considerada demasiado 'árida' para los principiantes, e incluso abrumadora, se debe a que, sin embargo, es la mejor a la hora de configurar correctamente y de forma automática todos los paquetes que uno se instala en el sistema, cosa que resulta realmente útil para alguien que no tiene aún conocimientos sobre Linux y espera no tener que pasarse horas y horas frente la máquina configurando el software.

Bien, no es la más rápida, pero desde luego, es la mas completa y la más conseguida. Cabe destacar que ya está en la calle la versión 2.1 de esta distribución, que incluye como opción otra manejador de paquetes distinto al *dselect* aquí comentado, el *apt*.

A.1 ¿Por dónde empiezo?

Bien, ahora queremos instalarnos la última distribución que hemos conseguido, en este caso la Debian 2.0 (código clave hamm) que se distribuyó gratuitamente junto con el número 4 de la revista Linux Actual, y que, por sujetarse a la GPL, evita que incurramos en delito alguno instalándonosla del CD que nos haya dejado un amigo.

¿Qué queremos hacer?, bien lo más probable es que usemos Windows 9x para todo, incluso INET si tenemos, y queramos conservarlo (aunque sólo sea por un tiempo hasta que migremos TODO a Linux), así que, vamos a instalar Linux en una partición distinta que vamos a hacer en nuestro disco duro.

A.1.1 ¿Qué es una partición? ¿Cómo hago yo una?

Buena pregunta. Una partición es básicamente un coto independiente dentro de nuestro disco duro. Luego, a esa partición le podemos dar el formato que queramos (FAT, FAT32, HPFS, EXT2..) para guardar los archivos.

Linux necesita de una/dos particiones exclusivas para él, una para el SO y otra para usar de zona de intercambio de memoria a disco (swap). Si tienes más de 64Mb de RAM, podrías pasar de partición swap. Sin embargo, por motivos de rendimiento en los que no vamos a entrar, siempre es recomendable tener algo de swap, aunque sea poca. Para ello hemos de cambiar la partición primaria de nuestro disco, en el cual tendremos sólo una típicamente con el formato FAT32 para el Windows, para reducirla y añadir otras dos primarias, la de Linux y la de swap.

¿Y como lo hago?, tenemos dos opciones a la hora de reparticionar un disco:

De nuevas.. “format c:“

Básicamente, consiste en formatear el disco duro, y crear las particiones de cero, para luego reinstalar, primero el Windows, de un backup, quizá, o de nuevas, recuperando nuestros archivos de un backup, y luego ponernos con la instalación de Linux.

Es importante hacerlo en este orden, pues Windows reescribe el MBR (master Boot Record) de nuestro duro, que también necesitaremos para Linux y el Loader del SO (el LILO, LInux LOader), que será el encargado de elegir que SO arrancamos.

Esta opción a veces resulta más que recomendable, sobre todo cuando el Windows lleva un tiempo haciendo 'cosas raras' y pidiendo a gritos una de sus reinstalaciones.. :)

Denial Of Service Attacks son ataques destinados a explotar un bug o fallo del sistema para conseguir paralizar el sistema y que deje de funcionar.

Ante todo este tipo de amenazas, no nos queda más remedio que ser cautelosos y saber en todo momento qué es lo que vamos a introducir en nuestro sistema y de donde proviene. Cualquier descuido podría conllevar la posible entrada en el sistema de alguien indeseado o la pérdida parcial o total de nuestra información (otra buena razón para hacer copias de seguridad).

8.8 Últimos consejos

Aunque lógicamente sólo hemos podido tratar una minúscula parte de la seguridad de sistemas informáticos, el lector debería investigar un poco por su cuenta y mantenerse al día. No hay que olvidar que cada día se descubren nuevos fallos en los programas y si uno no se mantiene actualizado es fácil que se le pasen ciertas cosas.

Es conveniente asimismo que el lector pruebe la seguridad de su sistema con programas que busque vulnerabilidades en su máquina o red, dichos programas se encuentran en Internet y son de libre distribución, entre ellos recomendaríamos: Tiger, Nessus, IIS (Internet Security Scanner), etc.

Y sobre todo hay un consejo fundamental: *mantente paranoico*.

1.3.3 S.u.S.E.

S.u.S.E. es una empresa alemana dedicada a Linux desde varios campos, como mediante el desarrollo de una suite comercial para linux, llamada Aplixware, conjuntamente con Red Hat, que posee su propia distribución. Su actual versión es la 6.2, que ya incorpora las glibc. <http://www.suse.de>

1.3.4 RedHat Eurielec

Eurielec Linux es una distribución basada en RedHat, creada por Eurielec, un club de estudiantes de la ETSI de Telecomunicaciones de Madrid (UPM).

Es una distribución pensada especialmente para los hispano-parlantes. Tiene la instalación traducida al castellano, así como numerosos programas y toda la documentación sobre linux de LuCAS (Linux en CASTellano), las man en español, scripts, etc. <http://www.etsit.upm.es/~eurielec/linux>

1.3.5 Otras

Existen, además, muchas otras distribuciones de Linux, como puedan ser la propia de Caldera, ó Slackware, la más difundida años atrás, pero que se ha abandonado por la imposibilidad por parte del individual que la compilaba de añadir los cambios que pedían la comunidad Linux, como un sistema de paquetes, similar al RPM de RedHat y S.u.S.E. ó al Dselect (dpkg) de Debian.

1.4 Variantes

Como el único estándar que existe en Linux es, al fin y al cabo, el kernel, lo que hay son estándares de facto, como lo pueden ser el compilador de C y C++ de la GNU, el gcc (ampliamente usado en entornos Unix), o el intérprete de comandos, la bash. Pero, de todas formas aparecen las diferencias. Básicamente cogen las aplicaciones del proyecto GNU y las usan en Linux. La más destacable es la de las dos versiones de librerías de C que coexisten en las distintas distribuciones. La libc5, y la glibc ó libc6, que es un desarrollo de la GNU incompatible con las libc5, pero que implementa una serie de mejoras sustanciales para el programador. Si bien ambas pueden convivir y conviven en una misma máquina Linux, progresivamente se está imponiendo la glibc. Por ejemplo, las últimas versiones de Red Hat y Debian (las 5.x y la 2.x, respectivamente) implementan las glibc (aunque cabe destacar que en la Debian todos los paquetes están recompilados con ésta última, no siendo así en RedHat), mientras que de momento los usuarios de S.u.S.E. que quieran actualizarse han de actualizarse a la versión 6.0.

1.5 Características de Linux como Sistema Operativo

Pero, ¿puedo usarlo en mi PC?.. un poco de palabrería técnica tampoco vendrá mal³:

- **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preeventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- **Multiusuario:** Muchos usuarios usando la misma máquina al mismo tiempo.
- **Multiplataforma:** Las plataformas en las que en un principio se puede utilizar Linux son 386, 486, Pentium, Pentium Pro, Pentium II, AMD Kx, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.
- **Multiprocesador:** Soporte para sistemas con más de un procesador está disponible para Intel y SPARC (hasta 16 cpu's).
- **Funciona en modo protegido 386.**
- **Protección de la memoria entre procesos,** de manera que uno de ellos no pueda colgar el sistema.

³fuente: INFOSHEET-Como, de Iván Casado

- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: A una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha. Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio. Este límite se puede aumentar fácilmente con el cambio de unas cuantas líneas en el código fuente.
- La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.
- Librerías compartidas de carga dinámica (DLL's) y librerías estáticas.
- Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.
- Compatible con POSIX, System V y BSD a nivel fuente.
- Emulación de iBCS2, casi completamente compatible con SCO, SVR3 y SVR4 a nivel binario.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Control de tareas POSIX.
- Pseudo-terminales (pty's).
- Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si el ordenador ya tiene una FPU (unidad de coma flotante), esta será usada en lugar de la emulación, pudiendo incluso compilar tu propio kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.
- Soporte multilinguaje de páginas de código siendo bastante fácil añadir nuevos dinámicamente.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 256 en la nueva serie del kernel.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud (Ext2).
- Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no es necesario ningún comando especial para usar la partición MS-DOS, esta parece un sistema de archivos normal de Unix (excepto por algunas restricciones en los nombres de archivo, permisos, y esas cosas). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro, pero sí lo son mediante el DOSEMU.
- Soporte para VFAT (WNT, Windows 95y Windows 98), y también para FAT32.
- Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.
- Soporte en sólo lectura de HPFS-2 del OS/2 2.1

8.5 Los registros del sistema (logs)

A menos que lo configuremos de otra manera, la mayor parte de los mensajes del sistema van a parar al directorio */var/log*, donde se guardan diferentes tipos de registro. Como cada distribución de Linux cambia la forma de almacenar los registros, los ficheros más usuales son:

messages Guarda todos los mensajes de la consola del sistema y otros mensajes de la utilidad *syslog*.

lastlog Guarda las fechas y horas de entrada de los usuarios y generalmente de los intentos fallidos.

xferlog Almacena los accesos por ftp.

sulog Archiva quien y cuando ha usado el comando *su*.

utmp Guarda registros de los usuarios que hay conectados en el momento.

Hay generalmente más fichero, pero estos suelen ser los más usuales. Además podríamos crearnos nosotros mismos nuestro fichero de logs donde acabaran los datos que nos interesan.

8.6 Eliminar servidores innecesarios

La mayoría de las veces nos encontraremos con que las distribuciones de Linux al arrancar hacen comenzar un montón de programas servidores. Lo más seguro es que no necesitemos la mayoría de ellos, así que lo más recomendable sería hacer que no se ejecutaran.

Para ello simplemente editaremos el archivo */etc/inetd.conf* y eliminaremos o comentaremos la línea que se refiere a ese servidor. Por ejemplo si quisiéramos dejar de tener un servidor para *finger*, tendríamos que eliminar la línea:

```
# /etc/inetd.conf
...
finger stream tcp nowait nobody /usr/sbin/tcpd in.fingerd -w
...
```

Lo ideal es eliminar todos los servidores que no vayamos a utilizar y aquellos que puedan proporcionar información al exterior sobre nosotros (pop3, finger, smtp, ...).

Además es muy buena idea eliminar los servidores de telnet y los comandos-r, y dejar en su lugar el ssh. También habría que plantearse la necesidad de tener un servidor ftp.

8.7 Posibles amenazas informáticas

Además de la posible entrada en nuestro ordenador o que haya alguien viendo pasar los paquetes que pasan, existen otro tipo de amenazas (en realidad puede haber tantas como se nos ocurra), pero las más habituales suelen ser:

Programas de test de seguridad y toolkits: son programas diseñados para testear la seguridad de un sistema, pero alguien malintencionado puede utilizarlos para descubrir vulnerabilidades en la configuración de la máquina.

Puertastraseras son “trampas” dejadas en los programas para permitir el acceso a alguien a tu sistema.

Bombaslógicas son programas maliciosos que se activan una vez se cumplen ciertas condiciones (se llega a una fecha determinada, un determinado usuario ejecuta un programa).

Virus programas que modifican copias de otros programas para copiarse en ellos (y quizás realizar acciones destructivas).

Gusanos programas perniciosos capaces de moverse en la red y copiarse de ordenador en ordenador sin necesidad de copiarse a sí mismo.

Troyanos son programas que generalmente se esconden en otros programas para pasar desapercibidos y realizar su función.

Conejos (o cualquier animal): su objetivo es agotar los recursos del sistema.

8.4.2 Uso del programa

Se ha comentado que la encriptación por clave pública usa dos claves, por lo tanto, lo primero será generarlas. Las claves no son frases que escribamos nosotros, ni tendremos que recordarlas ni teclearlas. Son números muy grandes (media pantalla) que se guardan en el disco duro. Basta con que no nos accedan al disco duro para que permanezca segura. Además, como las claves se intercambian cada vez que se conecta, no es necesario conservarla siempre, podemos crear una nueva cada vez que queramos. El comando para crearla es:

```
ssh-keygen
```

Que admite muchos parámetros, como el tamaño de clave (mayor tamaño más seguridad, pero más lento). El tamaño que viene por defecto será suficiente si no somos alguna persona perseguida por medio planeta, e incluso en esa situación se tardarían años en romper la clave.

La aplicación central es el cliente, esto es, el programa que nos permite conectar con una máquina que tenga instalado el servidor.

Se llama con:

```
ssh -l usuario máquina
```

Hay que darle el usuario que somos en la máquina remota. Si estamos en nuestro ordenador como root, y tenemos una cuenta como pepe, tenemos que decirlo, para que no nos confunda con el root. Tras hacer esto, y si la cosa funciona, nos pedirá la clave. Si alguien quiere ver más en detalle cómo funciona por dentro, puede incluir la opción `-v`. La clave ya se manda de forma encriptada por la red, con lo que nadie podría verla.

Y ya entraremos en modo encriptado, con un shell. Pero quizás no es un shell lo que queremos, quizás queremos exportar el display y poder cargar programas de las X Windows en el ordenador remoto y verlas en el nuestro ... No hay problema, el SSH lo hace por sí solo. Una vez conectados, y si estamos en las X, podremos hacer un "xcalc" y cargar la calculadora en nuestro display, por supuesto, de forma segura. Caso práctico: En la universidad, si el SSH estuviera instalado en el proxy de internet, podríamos cargar el SSH de forma remota, y nadie vería qué estamos visitando.

Sin embargo, falta algo ... con esto, hacemos un telnet, pero ¿cómo hacer otro tipo de conexiones? Para eso está la redirección de puertos.

Es un sistema que nos permite acceder a un puerto cualquiera de una máquina remota, engañando a nuestro programa. Por ejemplo, queremos mirar el correo en apolo, puerto 110. Habría que ejecutar el SSH así

```
ssh -f -S -l user -L 2000:apolo:110 apolo
```

Nos pediría la clave y no haría nada más ni entraremos en apolo ni nada. El `-f -S` es para que haga eso, que no haga nada, pero la conexión sigue abierta. Miramos una línea que pone 2000:apolo:110. Eso significa que estamos redirigiendo nuestro puerto 2000 al puerto 110 de apolo a través del SSH. Si ahora hacemos un telnet a nuestra máquina al puerto 2000, veremos cómo apolo nos saluda en el puerto SMTP. Estamos conectándonos de forma remota al puerto 110 por un canal seguro. Haciendo esto para cada puerto de internet como el de mandar correo (apolo:25), el proxy de internet (www:80), etc. podríamos acceder a todas las máquinas de forma segura.

Ahora sólo queda una cosa, que puedan acceder a nuestra máquina. Para eso, la aplicación se llama `/usr/local/sbin/sshd` simplemente ejecutándola ya se cargaría como cualquier otro demonio, y aceptaría conexiones. Para cargarlo siempre al entrar, habría que incluirlo en algún script de inicio.

No es recomendable incluirlo en el inetd, porque el SSH tarda unos segundos desde que se carga hasta que acepta conexiones, ya que tiene que generar las claves, y eso es incluso peligroso. Si nos hicieran 40 conexiones a la vez, habría 40 sshd en la memoria generando claves.

Los ficheros de configuración se encuentran en `/etc/ssh*`, y es recomendable mirarlos un poco, así como los manuales, puesto que se pueden hacer más cosas, como cambiar la encriptación, limitar los accesos, cambiar el nivel de seguridad (y velocidad), etc.

El SSH se puede obtener en www.ssh.org, y luego un enlace que hay al Secure Shell.

El comando *last*

Cada vez que un usuario sale o entra en el sistema, se almacena en el fichero `/var/log/wtmp`. Para comprobar las últimas entradas de alguien se acude al comando *last*. Su salida nos proporcionará información sobre sus accesos, de esta manera si tecleamos "*last root*" veremos los accesos del superusuario (generalmente produce una salida larga).

- Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM, incluyendo Microsoft Joliet Nivel 3.
- TCP/IP, incluyendo ftp, telnet, NFS, etc.
- Appletalk.
- Software cliente y servidor Netware.
- Lan Manager / Windows Native (SMB), software cliente y servidor.
- Diversos protocolos de red incluidos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc. AX.25 es el protocolo de red de las transmisiones de Radio Paquete, habitualmente conocido como Packet.
- Diseño modular del kernel, abandonando el uso de un kernel monolítico.

1.6 Últimas Tecnologías

En la última versión oficial del kernel estable, la serie del 2.2.xx, que va por el 2.2.5, se quiso dar soporte a las últimas tecnologías.

1.6.1 IrDA

El kernel implementa soporte para ésta nueva tecnología de puertos infrarojos, especialmente práctica para los ordenadores portátiles (exceptuando la tecnología de Toshiba, pues se ha negado a ayudar a su inclusión en el soporte del kernel).

1.6.2 DVD

Intención del grupo de desarrolladores de incluir soporte para los discos versátiles digitales, que no se ha visto cumplida por falta de programadores para dicho campo. Se prevee que se vaya incluyendo en las siguientes versiones estables del kernel, si bien aún no existen aplicaciones para dicha tecnología. Sin embargo, las grandes multinacionales están poniendo todos los impedimentos para ello. Ya que ellos no proporcionan un software para la reproducción de DVD's en Linux, un grupo de hackers creó de la nada un soft capaz de hacerlo, el DCSS. Sin embargo, esto ha llevado a la detención de un joven finlandés de 16 años miembro del grupo, y a una causa judicial en progreso.

1.6.3 USB

De forma similar, los desarrolladores del kernel han expresado su deseo de incluir soporte para ésta nueva tecnología de periféricos, pese a que aun no hay mucho código.

1.6.4 UDMA

Las nuevas normas de acceso a disco están ampliamente soportados por el kernel, si bien (como en mi caso) además hace falta que tu placa madre lo soporte. ;)

1.6.5 IPv6

El que será dentro de unos años el nuevo estándar en protocolos de INET ya presenta código experimental en el kernel estable.

- Cualquier tipo de nombre propio, el de un familiar o allegado (generalmente los diccionarios contemplan una larga lista de nombres).
- El nombre del sistema operativo que se utiliza (ej: Linux, Linuz, LiNux, ...).
- Ningún tipo de número de teléfono o fecha.
- Esquemas simples del teclado (ej: qwerty, 12345³, ...).
- Lugares geográficos o de sitios conocidos (ej: Melilla, bardepepe, ...).
- Palabras fácilmente relacionables con un hobby o afición (guru, wizard, gandalf, sexo, ...).
- Ninguno de los anteriores al revés.
- Ningunos de los anteriores con un dígito al final o antes.

Por lo general lo más eficaz suele ser elegir una combinación alfanumérica sin sentido. Con este método habremos parado a más de algún tipo que pretendiera crackear nuestro fichero de claves (aunque sigue siendo factible que lo consiga).

Porque todavía quedan posibilidades de que sea inseguro es buena idea cambiar los passwords a menudo, y para el superusuario esto debe ser una obligación. No hay que olvidar que además de podernos crackear el password un posible intruso puede sniffear el medio físico de comunicación para conseguir las claves, lo que nos lleva al siguiente tema.

8.4 Secure Shell (ssh)

El SSH (Secure SHell) es un programa que permite acceder a una máquina remota de forma segura. En las redes de tipo broadcast, como ethernet, donde muchas máquinas están conectadas al mismo cable, es muy sencillo monitorizar el canal y así poder ver lo que la gente está transmitiendo. Se podrían averiguar passwords, mensajes de correo, ver qué páginas web vemos, etc.

Este programa nos permite hacer una conexión remota (tipo telnet, en principio) a otra máquina en Linux o Unix de forma encriptada. El programa es gratuito para Unix en Linux, pero no lo es para otro tipo de sistemas operativos como Windows.

El principal problema de los sistemas de encriptación que se usaban tradicionalmente era la necesidad de intercambiar una clave por un medio seguro. Si la clave se transmite por la red, es igual de fácil capturarla y decodificar con ella los paquetes encriptados. Y llamar por teléfono a la persona, o ir a su casa, para poner la clave, no es una buena solución.

Pero ese problema está solucionado con las claves de tipo público. Aquí tenemos dos llaves, una pública que se puede leer, y una privada que tenemos que guardar. La pública sólo sirve para codificar datos, pero con ella no podemos volver a decodificarlos, eso lo hace la clave privada.

Este método permite una codificación segura, ya que al iniciar la conexión ambas máquinas se intercambian sus claves públicas, y a partir de ahí se empieza a encriptar. Aunque un ¡¡cotilla!! intercepte el las claves públicas, no le servirán de nada.

8.4.1 Instalación

La instalación del SSH es casi automática. Una vez descomprimido en un directorio, sólo tenemos que hacer:

```
./configure
```

que se encargará de configurar la compilación del SSH según nuestro sistema.

```
make
```

Que compilará el programa

```
make install
```

Que instalará los ejecutables y los manuales.

Con esto, ya tenemos instalado el SSH, que tiene varios programas.

³Sí, la clave que un tonto podría en sus maletas! :-)

- ¿De qué o quién lo tengo que proteger?
- ¿Cuánto tiempo, esfuerzo e incluso dinero voy a estar dispuesto a utilizar para conseguir una protección adecuada?

Una vez que conozcamos cuáles son nuestras necesidades ya sólo quedará obrar en consecuencia para cubrirlas, eligiendo entre diferentes soluciones.

Cuatro sencillos pasos para conseguir mayor seguridad

Conseguir un sistema seguro puede llegar a suponer un montón de trabajo. Si simplemente no tenemos tiempo para hacer un análisis detallado de de todo lo que podamos llegar a necesitar, recomendamos seguir estos pasos para al menos tener un nivel de seguridad decente:

Decide cómo de importante es la seguridad para tu sistema Si crees que es un asunto prioritario deberás dedicar una mayor atención y un estudio serio de los riesgos. Si por el contrario es algo que no te merece mayor importancia puedes dejar el sistema con los mecanismos que trae por defecto el sistema operativo (generalmente creemos que ésto es caer en un error).

Educa a tus usuarios La gente que vaya a utilizar tu sistema (familia, amigos, clientes, ...), deben conocer los peligros y riesgos que pueden tener una actitud demasiado peligrosa. Deben saber a quien acudir en caso de necesidad y conocer las bases de seguridad (tales como la privacidad de la contraseña).

Realiza un plan para realizar copias de seguridad Esto es algo fundamental para evitar desastres que puedan ocurrir. Los backups deberían realizarse con más o menos periodicidad en función del ritmo de actualización de la información. Nótese que el realizar estas copias también nos mantendrá preparados frente a posibles averías de hardware.

Sé curioso y paranoico No, no es broma. Mantenerse en un estado de tensión te hará cometer un montón de malentendidos, viendo intrusiones por todas partes, pero también te hará descubrir cosas que de otro modo te pasarían por alto. ¡Revisa los logs!

¿Es Unix un sistema seguro?

Es muy frecuente oír que Unix (y en consecuencia Linux) es un sistema operativo poco seguro. Esto no es más que una visión distorsionada de la realidad, ya que por tradición los únicos sistemas que estaban dotados de capacidad para trabajar en red eran los sistemas tipo Unix (mayoritariamente), así que no era posible entrar donde no había conexión.

En los últimos años se ha visto como esta creencia se desmoronaba. A medida que se popularizaron las redes para todos los sistemas, surgían formas de intrusión en todos ellos. Un claro ejemplo de todo es la ingente cantidad de programas “*troyanos*” que asaltan las plataformas basadas en Windows, y que pueden tener efectos desastrosos. En realidad no existe un sistema operativo más seguro que otro, todos son penetrables, ya que en todos ellos existen fallos de programación que hacen posible la vulnerabilidad del sistema.

Pero la gran ventaja de Unix/Linux en este aspecto es que nosotros como administradores podemos fijar el nivel de seguridad que queramos. De este modo podemos dejar el sistema con una simple protección basada en la autenticación login-password o, por el contrario, tener unas medidas de fortificación del sistema en múltiples niveles, tales como: múltiples passwords, comunicaciones encriptadas, comprobación de identidad del cliente, control de logs del sistema, etc. Pero para conocer que actitud debemos tomar deberemos elaborar una *política de seguridad*.

8.3 Claves de usuario

La seguridad del sistema comienza con lo más básico, que es como la gente accede a sus cuentas. La forma de acceder es la siguiente: cuando se crea un usuario, su password se almacena en el fichero */etc/passwd* o bien en el */etc/shadow* (es lo más habitual ultimamente, es lo que se conoce como *shadow password*). Las contraseñas se encuentran encriptadas con un algoritmo de un sólo sentido (no puede desenscriptarse)². De este modo cuando el usuario introduce su contraseña al entrar, ésta se codifica y se compara el resultado codificado, y si coinciden se permite el acceso.

Es de importancia vital el elegir buenos passwords, ya que existen métodos de fuerza bruta que a través de diccionarios especiales comprueban si el usuario ha elegido una contraseña fácil de adivinar. Asimismo hay que recordar que el fichero que contenga las contraseñas debe ser secreto. Una buena política para la elección de passwords descartaría los siguientes tipos de contraseñas:

²A menos que esté dispuesto a gastarse unos 250.000 dólares en una máquina especial que sólo sirve para ese objetivo.

Capítulo 2

Comandos Básicos

‘..Y toma de contacto’, debería llamarse ésta sección. La mayoría de los comandos, como es normal, y en especial en el caso de los más básicos, son los comandos típicos de todo sistemas Unix, aunque, en general, con mayor manejabilidad, potencia, y posibilidades de uso. Por ejemplo, el comando de copiar en el DOS es *copy*, ¿nó?.. pues en Linux es *cp*.

2.1 Ayuda en pantalla

Linux, como Unix, implementa las llamadas ‘páginas de manual’, que se invocan con *man* ‘*comando*’, y presentan en pantalla una descripción y parámetros del comando en cuestión. Además, aquellas utilidades GNU (en general, las que empiecen por *g*) tienen una utilidad de ayuda específica análoga a la anterior llamada *info*.

2.2 El sistema de Archivos de Linux

2.2.1 Permisos de los Archivos

Dado que Linux es un sistema multiusuario enfocado a redes, el sistema de archivos de Linux provee de un sistema de permisos de archivos para evitar manipulación por otros usuarios de los archivos de uno. Asimismo, este sistema dispone de modos para que unos archivos sean compartidos por un mismo grupo de usuarios. Por ejemplo, en la mayoría de los sistemas Linux con más de un usuario, y un acceso a Internet no general, quizás interese crear un grupo llamado módem que posea permisos para usar el mismo. Los permisos de un archivo se pueden ver mediante la ejecución del comando de listado, *ls*, con un modificador para que la lista sea larga: *ls -l*

```
-rw-r--r-- 1 raxi users 3147 Dec 26 1993 README
```

En este ejemplo el dueño del fichero que se llama README es raxi. El grupo de usuarios al que pertenece es users. La fecha de su creación es el 26 de diciembre de 1993. El tamaño es de 3147 bytes. Las 10 letras y guiones iniciales son los permisos y se interpretan de la siguiente forma:

```
(dir) (rd,wd,ed) (rg,wg,eg) (ro,wo,eo)
```

(dir): “d” si es un directorio, “l” si es un enlace, “b” si es un dispositivo de bloque, “c” un dispositivo de caracteres, “p” si es una tubería, y “-” si es un archivo ordinario

(rd,wd,ed): valen “r”, “w”, “x” si el dueño tiene permiso de lectura, escritura o ejecución sobre el fichero, respectivamente.

(rg,wg,eg): valen “r”, “w”, “x” si el grupo tiene permiso de lectura, escritura o ejecución sobre el fichero, respectivamente.

(ro,wo,eo): valen “r”, “w”, “x” si los demás tiene permiso de lectura, escritura o ejecución sobre el fichero, respectivamente.

Un guión en cualquier posición indica que no se tiene el permiso correspondiente.

Permisos especiales

Existen cuatro permisos especiales, que son los siguientes:

- SGID
- SUID
- Sticky Bit (bit de permanencia)
- Locking Bit (bit de Cierre)

En realidad no suponen ninguna adición, sino que son casos especiales de los permisos normales.

SUID: El SUID significa 'Set Users ID', o sea, fija la identificación del dueño y el programa se ejecuta con la identificación y los permisos del dueño. Un caso típico es el del programa *passwd*, que cambia la contraseña de entrada de los usuarios modificando el archivo */etc/passwd*. Para que un usuario normal no pueda modificar el archivo */etc/passwd*, se activa el SUID al programa *passwd* y este, como se ejecuta con los permisos root, sí puede modificar el archivo, pero sin peligro de que un usuario lo haga.

SGID: El SGID es análogo y significa 'Set Group ID', es decir, 'Fijar la Identidad del grupo'. Lo que provoca es que cuando un usuario ejecute un programa con el SGID activo, este se va a ejecutar con la identidad (y se supone, los privilegios) del grupo al que pertenece el archivo.

Locking Bit (bit de Cierre): Este es el mismo bit que el de SGID, pero cuando no tiene permisos de ejecución de grupo el archivo sobre el que se actúa. Lo que hace es permitir que un programa cierre el acceso a un archivo evitando así escrituras simultáneas que conlleven una corrupción del fichero.

Sticky Bit (bit de permanencia): Este bit, al contrario que los otros, en los que lo pueden activar el dueño, o cualquiera con permisos para ello, sólo lo puede activar root. Provoca que el programa al que se le aplica quede residente en memoria de forma que la próxima vez que sea llamado su carga sea más rápida. Cuando se le aplica a un directorio, impide que nadie más que su dueño puede renombrar o borrar ningún archivo del mismo.

Interdependencia de los Permisos

Los permisos de un archivo, a su vez, dependen de los permisos del árbol de directorios en el que están. Por ejemplo, si tenemos un archivo llamado *cosa* en el directorio */home/raxi*, con los siguientes permisos:

```
[raxi@rax.uax.es /~]$ ls -l
total 3
-rwxrwxrwx 1 raxi users 7 ene 9 18:26 cosa
```

donde *~* significa el directorio casa del usuario (en este caso */home/raxi*), y el directorio *raxi* tiene éstos otros:

```
[raxi@rax.uax.es /home]$ ls -l
total 3
drwx----- 39 raxi users 3072 ene 9 18:26 raxi
```

La consecuencia directa es que el archivo *cosa*, de hecho, sólo es legible para el propio usuario *raxi*, ni siquiera lo es para los de su mismo grupo (*users*).

2.2.2 Dispositivos

En unix, todo dispositivo (pantalla, disco duro, teclado, impresoras.. etc) tiene 'forma' de un archivo ó archivos. Es decir, todo dispositivo tiene asociado uno ó más archivos.

Estos, normalmente, se encuentran en el subdirectorio */dev* del árbol del sistema de archivos. Así, por ejemplo, el archivo */dev/lp1* hace referencia a la impresora que tenemos conectada al primer puerto paralelo y */dev/ttyS0* al primer puerto serie (¿el módem?). Cuando un programa intenta ejecutar ó acceder a ese archivo es el kernel el que se ocupa de llamar a un driver para el mismo.

Existen dos tipos esenciales de dispositivo en función de su forma de transferir los datos:

Capítulo 8

Seguridad del sistema

La seguridad informática es algo que ha ido adquiriendo cada vez más relevancia a medida que el uso de las redes de ordenadores se han convertido en algo cada vez más mayoritario. Entre tanta gente conectada a la Red siempre surgen colectivos e individuos que atentan contra la integridad y la privacidad de los sistemas, ya sea por razones económicas o por puro reto intelectual. De cualquier forma, la amenaza es real y si nos interesa tener nuestros contenidos a salvo tendremos que tomar medidas para evitar la intrusión en nuestros sistemas. En este capítulo detallaremos algunas reglas fundamentales que sirvan como base para conseguir un entorno seguro.

8.1 Introducción a la seguridad

Cuando nos referimos a la expresión “*seguridad informática*” no estamos sólo comentando la forma de conseguir que alguien acceda a través de Internet a nuestro ordenador, sino que hablamos de un conjunto de actitudes y acciones destinadas a varias cosas:

- Garantizar la confidencialidad de nuestra información.
- Asegurar que nuestras conexiones no sean espiadas o atrapadas¹.
- Que podamos estar seguros de que la información que tenemos no haya sido modificada.
- Conseguir que nuestra identidad no pueda ser falsificada y utilizada por otros.
- Evitar los problemas que conllevarían una pérdida de datos o incluso el borrado de nuestra información.

Estos son algunos de los objetivos que pretendemos conseguir, para ello utilizaremos distintas herramientas y sobre todo deberemos tener una forma activa de enfocar el tema de la seguridad. Esto es algo muy importante, ya que la seguridad del sistema no es algo que se haga solo, necesita de nuestra supervisión.

8.2 Planificando las necesidades de seguridad

El binomio seguridad-accesibilidad siempre se encuentra enfrentado entre sí. Siempre que queremos hacer un sistema más seguro aumentamos la dificultad para acceder a él y viceversa. Para conseguir el punto de equilibrio que necesitamos debemos plantearnos antes de nada varios puntos, y una vez resueltos todo desembocará en la creación de una *política de seguridad* que nos permitirá ser más eficientes que dejándolo todo desordenado.

8.2.1 Conocimiento de los riesgos

El primer paso para mejorar la seguridad del sistema es contestar estas tres preguntas:

- ¿Qué es lo que intento proteger?

¹Es lo que se conoce como *hijacking*.

1. de carácter
2. de bloque

Como ejemplo de los dispositivos de bloque se encuentran todos los discos duros. Los discos duros, por ejemplo, en función del canal IDE en el que se encuentren, tendrán un dispositivo asociado. Por ejemplo, un disco duro en el master del segundo canal tiene asociado un archivo o archivos de la forma `/dev/hdc?` donde la interrogación corresponde al número de la partición (por ejemplo, `/dev/hdc1`).

Montar sistemas de archivos

Un caso especial, debido al sistema de archivos de Linux, aparece cuando queremos acceder a un dispositivo de almacenamiento removible típico, como un cd-rom ó un disquette. Estos presentan asociados cada uno un archivo en `/dev`. Para poder leerlos hemos de anejar a la estructura de directorio, añadiendo sus contenidos a ella. Esto se hace con el comando `mount` y a partir de las especificaciones que aparezcan en el fichero `/etc/fstab` que indican el directorio del árbol a donde hemos de añadir sus contenidos. Sin embargo, la nueva serie estable del kernel, la 2.2.x, implementa un 'automounter' para que sea el kernel el que realiza esta tarea cada vez que nosotros accedamos al directorio de montajes de un dispositivo (como el `/mnt/cdrom`)

2.2.3 El Árbol de Directorio

A continuación describiremos el típico árbol de directorio de un sistema Linux y sus convenciones:

/bin: Contiene los archivos ejecutables disponibles para todos los usuarios parte de la distribución base.

/sbin: Contiene archivos ejecutables parte de la distribución base exclusivos del superusuario.

/dev: Todos los enlaces a los dispositivos

/etc: Todos los archivos de configuración de la mayoría de los programas, así como los scripts de inicio del sistema.

/lib: Contiene las librerías (ó bibliotecas de funciones) de subprogramas compilados en C que pueden ser llamadas por programas C y enlazadas con ellas.

/usr Este directorio contiene a su vez otro arbol estándar propio:

bin binarios que no son parte de la distribución base

X11R6 Contiene los archivos de la implementación de las X Window de la XFree86 ¹<http://www.xfree86.org>

include cabeceras de las librerías

lib librerías

local subdirectorio para programas adicionales

games pues eso

sbin semejante a /sbin

/home Contiene los directorios casa de los usuarios del sistema

/opt semejante a /usr/local contiene programas adicionales (como las KDE, por ejemplo)

/mnt punto sugerido para montar dispositivos removibles (com /mnt/cdrom, por ejemplo)

/var Para almacenar archivos que varían su contenido con el transcurso del tiempo, como spools de impresoras

/proc supone un 'tipo de archivos' específico

/tmp para las copias temporales

2.3 Intérprete de Comandos (BASH - Bourne Again Shell)

Aquellos fanáticos del DOS, que seguro que habrá por aquí, estarán familiarizados con éste término, pues uno de los máximos exponentes de éste tipo de programas es el 4DOS. En el mundo Unix, en general, han existido varios intérpretes de comandos Unix populares. Entre ellos se pueden encontrar el sh, el ksh, ash ó el bash. Todos ellos existen en versión Linux, si bien, tanto en Linux como en Unix, es más extendido, y más completo es la bash. El intérprete de comandos es aquél programa que te llama los comandos del S.O. cuando tu se lo dices. Una posibilidad muy potente de los intérpretes de comandos Unix es la posibilidad de crear ficheros nido, ó batch (equivalente a los .bat de DOS²), donde se explicitan una secuencia de comandos que se realizarán al invocarse (con permisos de ejecución) ese fichero, también llamado script. Como ejemplo, ós pongo el script que tengo en mi sistema para automatizar la llamada a mi ISP y recoger mi correo

```
#!/bin/bash
#usr/bin/llamadaISP
clear
if [ -e /var/lock/LCK..ttyS0 ]; then
    echo parece que ya estás conectado
    echo entonces no hago nada.. y paso a recoger el correo
else
    echo Llamando al proveedor...
    pon
    echo esperando 50 segundos...
    sleep 50
    echo comprobando la conexión..
fi
if [ -e /var/lock/LCK..ttyS0 ];then
    echo parece que hay una conexión válida
    echo recogiendo el correo...
    fetchmail
    echo dandole tiempo
    sleep 2
    echo colgando...
    poff
    echo dandole tiempo a limpiar la cola
    sleep 5
    mutt -y
    exit
fi
echo No parece que haya funcionado...
exit
```

Como véis el fichero empieza con un comentario (con la almohadilla delante) que define el intérprete para el que está escrito. La siguiente línea es el nombre del archivo, y luego ya vienen los comandos. Pero no sólo se queda ahí, pues la bash implementa todas la secuencias de control típicas de la programación imperativa, como pueden ser los: for, while, repeat.. etc Mediante ello, podemos realizar programillas de un grado medio de complejidad. De aquí en adelante, todos los comandos que aparezcan se referiran a la bash, si bien la mayoría son comunes a todas las shells.

2.3.1 Archivos de configuración

Existen una multitud de archivos de configuración para la BASH, de los cuales el más usado es el .bashrc que se guarda en el directorio ~ de cada usuario. El . indica que es oculto (para ver los archivos ocultos basta ejecutar un *ls -a*). En él se definen variables de entorno como el PATH. Este es mi ~/.bashrc

```
# ~/.bashrc: ejecutado por bash(1) para shells no de entrada
```

```
#Alias y funciones específicas del usuario
PATH="$PATH:~/bin:/usr/games:/usr/local/bin"
```

²en realidad el sentido de la equivalencia es el contrario, :-)

7.6 IDE, Entornos de Desarrollo Integrados

Existen algunos programas que integran en uno único todas las fases de desarrollo, con editores, debuggers, compiladores, etc ...

En el mundo Linux, esto es poco común, ya que la gente que programa suele estar muy predeterminada a usar el editor que lleva usando toda su vida (especialmente entre usuarios de Emacs/Vi³). Sin embargo, estan empezando a aparecer.

Un ejemplo es el Visualage de IBM para programación en Java. Sigue la Filosofía del Glade, Visual Basic, etc.. a la hora de ayudarte a crear las interfaces gráficas para tu programa de forma sencilla y rápida. Sin embargo, otros dirán que con Emacs/Vim ya les vale, y no les hacen falta.

³yo uso vim, por cierto

7.3.1 Ejemplo. Programación en Java

Si repetimos el mismo ejemplo pero en Java (para consola): El código fuente:

```
//Hola.java
import java.lang.*;

public class Hola {
    static void main (String args[]){
        System.out.println("\n\tHola Mundo!\n");
    }
}
```

Ahora lo compilamos mediante

```
[raxi@rax.uax.es ~]\$ javac Hola.java
```

Y luego lo ejecutamos

```
[raxi@rax.uax.es ~]{1}\$ java Hola
    Hola Mundo!
[raxi@rax.uax.es ~]\$
```

La programación en Java, como ya sabréis, es especialmente explotada en applets (*aplicacioncillas*) asociadas a páginas web. Pero esto es sólo un ejemplo para ver cómo se programa en Linux.

7.4 GTK y el GLADE

Las GTK son las librerías que crearon los autores del GIMP (GNU Image Manipulation Program)² para programar el mismo, y que se han adoptado para el proyecto del gestor de escritorio ‘oficial’ de la GNU, el GNOME.

El glade es un programa que facilita la programación en GTK haciéndonos él el código correspondiente a la interfaz gráfica que vayamos a usar, con lo que luego sólo hemos de programar las acciones.

7.5 Compilar paquetes bajados de Internet

Muy a menudo, uno necesita una aplicación específica para Linux, y se va a Internet a buscarla. Normalmente, uno a encuentra en formato .tgz con las fuentes sin compilar.

¿Qué ha de hacer uno para instalarla? Bien, pues, supongamos que el paquete se llama *Aplic.tgz*. Primero hay que descomprimirlo. Para ello usamos el comando tar, de la siguiente forma:

```
[raxi@rax.uax.es ~]\$ tar xvfvz Aplic.tgz
```

Esto nos descomprime el paquete creándonos, por ejemplo, un directorio llamado *Aplic* que contiene las fuentes. En él, encontraremos un archivo *README*, o *INSTALL* con las instrucciones.

Básicamente se reducen a tres pasos:

- ejecutar un script predefinido que configure la forma de compilación (el Makefile, vamos). Se ejecuta con:

```
[raxi@rax.uax.es Aplic]\$ ./configure
```

- ejecutar la compilación propiamente dicha, mediante

```
[raxi@rax.uax.es ~]\$ make
```

- Instalar los binarios y las librerías necesarias para la aplicación: (para lo cual hace falta ser root)

```
[raxi@rax.uax.es ~]\$ make install
```

²Según ellos por que no les gustaban las Motif

```
CLASSPATH="$CLASSPATH:/usr/local/java/lib"
EDITOR="/usr/bin/vi"
```

```
export PATH CLASSPATH EDITOR
```

```
#lee el archivo de configuración global si existe
if [ -e /etc/bashrc ]; then
    . /etc/bashrc
fi
```

```
alias correo='mutt -y'
alias llama='/usr/bin/llamadaISP'
```

```
#establece los permisos con los que se crean por defecto los archivos
umask 022
LC_MESSAGES=es_ES
LC_ALL=es_ES
LANG=es
LESSCHARSET=latin1
export LESSCHARSET LANG LC_ALL LC_MESSAGES
```

2.4 Comandos más usuales

Brevemente, listaré los más comunes y su función. Sino nos podríamos estar aquí horas. Para más información, ya sabéis, RTFM³:

cd: Sirve para cambiar de directorio.

ls: lista el contenido del árbol

pwd: muestra la ruta absoluta del directorio actual

passwd: cambia nuestra clave de acceso

rm: Borra archivos

rmdir: borra directorios vacíos. Para borrar directorios llenos usar rm con el flag -r de recursividad.

grep “cadena”: comando muy útil en combinación con otros como cat y ls. Extrae ls cadena de la fuente

cat: similar al type de dos, muestra el contenido de un archivo por pantalla

find: buscador muy potente, desde archivos por nombre, hasta por fecha de modificación, de último acceso.. etc.

chmod: cambia los permisos de un archivo. Por ejemplo chmod u+x le da el permiso de ejecución al dueño del archivo.

chown: cambia el dueño y el grupo de un archivo. (chgrp)

ln: crea enlaces entre archivos (tanto simbólicos como ‘duros’)

cp: copia archivos

mv: mueve archivos

lpr <archivo>: imprime el archivo en la impresora predeterminada.

gzip/gunzip: comprime archivos.

³Este breve acrónimo suele ser habitual en los foros de discusión de Linux cuando se preguntan obviedades, o no tienes ni p*ta idea de qué contestar, y le quieres tomar el pelo a quien te hizo la pregunta. Existen dos acepciones, la políticamente correcta, y la que no. La que no dice que significa ‘Read The Fucking Manual’ (léete el jodido manual).. de la otra.. pues no me acuerdo, X’DDDD

bzip/bunzip: lo mismo, pero mediante el algoritmo bzip2

tar: crea un archivo de 'cinta' de uno ó mas archivos. En conjunción con un compresor crea los *.tgz y *.tar.gz

cal: muestra un calendario

date: muestra la fecha y la hora actual

df: muestra el uso del disco

free: muestra el uso de la memoria

du: lista el tamaño en bytes.

vi: llama al editor vi

view: llama al editor vi sin derechos de escritura

exit: sales del sistema

logout: lo mismo.

```
[raxi@rax.uax.es ~]$ gcc -v hola.c -o hola
18:46:35Reading specs from /usr/lib/gcc-lib/i486-linux/2.7.2.3/specs
gcc version 2.7.2.3
/usr/lib/gcc-lib/i486-linux/2.7.2.3/cpp -lang-c -v -undef -D_GNUC_=2
-D_GNUC_MINOR_=7 -D_ELF__ -Dunix -Di386 -Dlinux -D_ELF__ -D_unix__
-D_i386__ -D_linux__ -D_unix__ -D_i386__ -D_linux__ -Asystem(unix)
-Asystem(posix) -Acpu(i386) -Amachine(i386) hola.c /tmp/cca07016
GNU CPP version 2.7.2.3 (i386 Linux/ELF)
#include "... search starts here:
#include <...> search starts here:
/usr/local/include
/usr/i486-linux/include
/usr/lib/gcc-lib/i486-linux/2.7.2.3/include
/usr/include
End of search list.
/usr/lib/gcc-lib/i486-linux/2.7.2.3/cc1 /tmp/cca07016 -quiet
-dumpbase hola.c -version -o /tmp/ccb07016
GNU C version 2.7.2.3 (i386 Linux/ELF) compiled by GNU C version 2.7.2.3.
as -V -Qy /tmp/ccb07016 -o /tmp/ccc07016
GNU assembler version 2.9.1 (i486-linux), using BFD version 2.9.1.0.19
ld -m elf_i386 -dynamic-linker /lib/ld-linux.so.2 -o hol
/usr/lib/crt1.o /usr/lib/crti.o
/usr/lib/gcc-lib/i486-linux/2.7.2.3/crtbegin.o
-L/usr/lib/gcc-lib/i486-linux/2.7.2.3 /tmp/ccc07016 -lgcc -lc -lgcc
/usr/lib/gcc-lib/i486-linux/2.7.2.3/crtend.o /usr/lib/crtn.o
[raxi@rax.uax.es ~]$
```

Como vemos, aparecen explícitas las cuatro fases de compilación que ejecuta el gcc por nosotros, buscando cabeceras del kernel, enlazando código, etc ...

7.2.4 Make

La herramienta make es una herramienta estándar en todo sistema Unix. Permite efectuar una compilación inteligente de programas, evitando compilar lo superfluo. Para ello se basa en la sintaxis específica de un archivo llamado Makefile, que configura la forma de actuar de make, a la hora de llamar al compilador, las opciones, etc.. Generalmente se usa para compilar código C.

7.2.5 Formato de los ejecutables

Linux soporta dos formatos de ejecutables, el a.out, que fue el primero, y el ELF (Executable and Linking Format). a.out (Assembled OUTput) es el formato de las primeras versiones tanto de Linux como de Unix. Sin embargo, hacia 1994, el equipo de desarrolladores de Linux decidió migrar de formato hacia el ELF, pues es mucho más flexible y manejable que el a.out. Este formato fue inicialmente desarrollado y diseñado por el USL (Unix System Laboratories), y se utiliza en los sistemas Unix del tipo SVR4 y Solaris 2.x

7.3 Programación en Java en Linux

Para hacer programas en Java, se puede bajar uno las JDK portadas a Linux (<http://www.blackdown.org>). Además, Sun ya está dando apoyo a la misma Blackdown para portar el nuevo Java, 2.0, a Linux. Hasta ahora, las JDK eran no-oficiales, pero reconocidas por Sun.

Una vez que nos bajemos de Internet (o consigamos) las 12M de las JDK, y las descomprimamos en el directorio /usr/local/java, sólo tendremos que incluir en nuestras variables (los archivos ~/.bashrc) PATH la dirección de los binarios y de las librerías y la API (classes.zip), y ya podremos programar en Java.

7.2.1 El gcc

El gcc es el compilador de la GNU para C, básicamente, aunque soporta varios lenguajes, como el ANSI C, el K&R C, el C++ y el Objective C. El gcc implementa las cuatro fases de la compilación. Además existe una versión extendida (egcs) que tiene capacidad para compilar en otros lenguajes como Fortran 77, C++ ...

Fases de compilación

Las fases son cuatro, a partir del código, ejemplo.c, se genera el ejecutable (en principio, a.out, aunque será mejor compilar con formato ELF):

Preprocesador (cpp) → C completo

Compilador (cc1) → Programa en ensamblador

Ensamblador (as) → Código objeto

Enlazador (o linkador¹) → ejecutable (a.out ó ejemplo)

Los nombres entre paréntesis son los de las herramientas encargadas de hacer cada proceso. Sin embargo, si ejecutamos gcc, éste realizará todas las fases en una sola. Si ejecutamos el gcc con la opción -v, recibiremos en pantalla información de las cuatro fases de compilación.

7.2.2 El gdb

Para todo desarrollo, siempre resulta útil la existencia de un 'debugger' (matabichos, literalmente), para poder evaluar paso a paso todas las instrucciones del programa.

Para poder usar el gdb con el programa compilado en C hará falta compilarlo con la opción -g del gcc, pues el gdb necesita algo más de información que se genera en un fichero aparte.

7.2.3 Ejemplo. Programación en C

Como ejemplo de programación en C, vamos a compilar en nuestra máquina el ya requetetípico programa de 'Hola Mundo'. Para ello necesitaremos tener en nuestro sistema el gcc (el compilador GNU de C), las cabeceras del kernel, así como quizá alguna que otra librería específica (como las ncurses para el manejo de terminales). El código es el siguiente

```
//hola.c
#include <stdio.h>

main()
{
    printf("\n\tHola Mundo!\n");
}
```

Para compilarlo hemos de ejecutar la siguiente orden:

```
[raxi@rax.uax.es ~]$ gcc hola.c -o hola
```

donde el flag -o indica el nombre del archivo binario ejecutable que queremos que genere el compilador. Una vez hecho, basta con teclear hola para ver lo siguiente:

```
[raxi@rax.uax.es ~]$ hola

    Hola Mundo!
```

```
[raxi@rax.uax.es ~]$
```

Si ejecutamos el gcc con el flag v para que sea 'verbose', la salida es la siguiente:

Capítulo 3

El Sistema XWindow

Este apartado "X" trata de hacer ver al usuario o usuaria en potencia de LINUX que puede ser (y de hecho es) un sistema tan "migable" y fácil de usar como lo puede ser cualquiera de los Sistemas de MicroSoft. NO necesitamos Windows (W) (a no ser que quieras jugar a esos juegos de "última generación" que a mucha gente gusta...) (Vamos a tener que portar los juegos a LINUX)¹ ;-GG X'DDDDD

Bueno, que lo sepas, que todo esto esta enfocado a hacer ver la facilidad, versatilidad, estabilidad del sistema de ventanas bajo LINUX. "X-Window"(sin la "\$").

3.1 ¿Qué son las Xwindow?

El sistema XWindow es un entorno gráfico para los sistemas Un*X. Tuvo sus inicios en el MIT. Las XWindow son un sistema de ventanas que permiten utilizar el ordenador de una forma fácil e intuitiva en modo gráfico mediante el ratón, iconos, ... Vamos, de forma similar al W95 pero sin tener que renunciar a la potencia de la máquina para poder utilizarla de forma cómoda. :-)"

En LINUX usaremos las XFree86, que no son más que una versión libre de las X del MIT, en su 11 versión y su distribución 6 (X11R6)

El proyecto XFree86 es una organización que produce una distribución gratuita de un Sistema de Ventanas que corren en sistemas UNIX (y OS/2). El proyecto XFree86 se ha basado tradicionalmente en plataformas basadas en Intel, pero en la actualidad se está empezando a soportar otras plataformas, como Alpha, ...

Los requisitos para correr XFree86 pasan por los requisitos mínimos para LINUX con una tarjeta de gráficos SVGA. Evidentemente, no se le pueden pedir peras al olmo y si ejecutamos las X en un 386 con 4 megas de RAM la cosa no irá del todo fina :-) El sistema X tiene un protocolo propio, independiente del SO, lo que da mucha flexibilidad de uso, potencia y seguridad.

Toda la información así como los distintos servidores se pueden encontrar en <http://www.XFree86.org>.

3.2 Instalando las X

3.2.1 Consiguiendo las X

Para instalar el sistema de ventanas X para LINUX tendrás que hacerte con los ficheros binarios del programa. Para ello puedes optar por compilarlos por tu cuenta o por conseguirlos ya compilados (lo que te puede ahorrar algunos quebraderos de cabeza). Los ficheros se pueden conseguir en MUCHOS servidores de FTP, así como en CDs de revistas y demás,... Un servidor típico de donde obtener las X es: <ftp.sunsite.unc.edu> donde encontrarás todo lo necesario en el directorio </pub/linux/X11>.

Tendrás que bajarte el fichero del servidor, (suelen tener nombres de chipset, o de tarjetas gráficas). Si no sabes que servidor elegir, te propongo que comiences instalando el de VGA16 o el de SVGA porque al ser más genéricos te darán menos problemas de instalación.

También necesitas otros muchos ficheros como son:

bin Con otros binarios del sistema.

¹Últimas noticias informan que la próxima versión del Civilization salió a la venta simultáneamente para MSDOS y LINUX!!

cfg Con los ficheros de configuración.

doc Con ficheros de documentación... esto es interesante porque es donde habrá que recurrir si algo falla.

inc Los ficheros "include"por si queremos compilar algo.

lib Las librerías necesarias.

fmt Las fuentes (las letras).

man Las páginas del manual.

...

Bueno, evidentemente los ficheros no se llaman así ;-) pero llevan esa "cadena"en su nombre... Los nombres varían según las versiones por lo que he decidido no restringirme a una versión en concreto. Hay otros MUCHOS ficheros que pueden ser necesarios, pero ya los irás echando en falta según los necesites }:-)

3.2.2 Ahora a instalar...

Esto es algo más complicado de explicar aquí... :-? Depende mucho de la distribución de LINUX que estés usando, de los paquetes que previamente hayas instalado... Bueno, en plan "general"podrás instalar en cualquier distribución sin muchos problemas los paquetes con extensión ".tar.gz"y leyendo el típico fichero "README" "INSTALL"... Y, dependiendo de la distribución, los paquetes *.rpm, *.deb aunque no es el formato del fichero un problema porque hay programas que convierten unos en otros y los instalan (man alien).

Una vez que te las has apañado para instalar todos los paquetes necesarios hay que lanzarse a configurarlo todo. Eso se hace de una forma muy cómoda con el programa "xf86config"que te permitirá configurar todo el sistema X, desde el ratón a la tarjeta de video, pasando por el monitor, ... Para llevar a cabo la configuración con éxito deberás conocer con exactitud el hardware que tienes en tu máquina... cosas como qué tarjeta gráfica tienes, la frecuencia de refresco de tu monitor, el tipo de ratón (ps2, ...)

Ese programa genera un fichero llamado "XF86Config"que es el fichero de configuración de las X y sobre el que luego, cuando algo te falle, tendrás que meterte a ver que puedes "retocar"y afinar tu configuración.

Otro programa muy práctico es "XF86Setup"que es en modo gráfico y te permitirá más o menos lo mismo que el xf86config pero con alguna otra mejora y más opciones. Yo te recomiendo que ejecutes los dos... y así los pruebas. :-)

Es importante que no pases por el programa de configuración sin leer las preguntas y contestar adecuadamente en función del hardware que tengas instalado en tu máquina porque de ello dependerá en buen funcionamiento del sistema.

3.2.3 "Castellanizando"las X

Ya tenemos las X funcionando!!... Bueno, más o menos, porque esta TODO en inglés, no puedes teclear las "Ñ's", ni los acentos de "mamá", "haré", ...

Esto es algo que suele dar bastantes quebraderos de cabeza, por lo que voy a tratar de echar una mano desde aquí. Estoy seguro de que me dejo algo, pero bueno. El sistema X lee la configuración del teclado cada vez que arranca de un fichero común para todos los usuarios o bien de un usuario particular para cada usuario. Esto es algo que tenemos que decidir a la hora de meternos a fondo con la configuración de las X,... tanto si el sistema está pensado para que lo use una sola persona como si está pensado para que lo usen varias siempre es mejor que se configuren las cosas de la forma más global posible.

Un fichero importante en este aspecto es el "xinitrc"... ¿qué dónde está? Pues depende de la distribución que uses... (prueba con "find / -name xinitrc") aunque en mi sistema, en estos momentos uso Debian 2.1) está en /etc/X11/xinit/xinitrc Echémosle un vistazo al "xinitrc"

```
#!/bin/sh
# $XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $
```

```
userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=LIBDIR/Xresources
sysmodmap=LIBDIR/Xmodmap
```

Capítulo 7

Entornos de Desarrollo

Linux era, hasta hace poco, un sistema operativo de hackers y para hackers, luego existen multitud de lenguajes de programación de alto nivel con los que se puede trabajar en Linux.

7.1 Lenguajes Soportados

Los lenguajes de programación soportados en Linux son muy numerosos, incluyendo los más comunes. Una breve lista puede incluir:

- C, C++, y Objective C
- Java
- Pascal
- Lisp
- Perl
- Occam
- S-R
- Gofer
- Fortran 77
- modula-2
- ...

En la lista, como nó, aparece el lenguaje C, con el que está escrito el kernel, además de lenguajes interpretados (lisp, Perl), seminterpretados (como el Java), ó funcionales (gofer), cubriéndose los tres campos de lenguajes de programación de alto nivel.

Como el C es la madre de Linux, le vamos a dedicar más tiempo. Pero antes hemos de hablar de las herramientas de desarrollo

7.2 Programación en C en linux

GNU, como ya hemos dicho, no es Linux. Linux es el kernel, exclusivamente. Las herramientas GNU son programas que están integrados en un proyecto más ambicioso que Linux, promovido por Richard Stallman (rms). Sin embargo, las herramientas de desarrollo del GNU son ampliamente usadas en Linux (desde el principio de linux), ya que son libres de uso y distribución.

Evidentemente aquí NO está todo el fichero, pero si alcanza a la parte que nos ocupa... poner las X en castellano. Aquí se ve donde están los xmodmap del sistema (general) o de cada usuario en concreto... /xmodmap Mi consejo es que probemos primero en un usuario y luego actualicemos el xmodmap del sistema una vez que todo funciona bien ;-)

Un Xmodmap que a mi funciona relativamente bien es el siguiente:

```
keycode 9 = Escape
keycode 10 = 1 exclam bar
keycode 11 = 2 quotedbl at
keycode 12 = 3 periodcentered numbersign
keycode 13 = 4 dollar asciitilde
keycode 14 = 5 percent
keycode 15 = 6 ampersand notsign
keycode 16 = 7 slash braceleft
keycode 17 = 8 parenleft bracketleft
keycode 18 = 9 parenright bracketright
keycode 19 = 0 equal braceright
keycode 20 = apostrophe question backslash
keycode 21 = exclamdown questiondown asciitilde
keycode 22 = BackSpace
keycode 23 = Tab
keycode 24 = q Q q
keycode 25 = w W w
keycode 26 = e E e
keycode 27 = r R r
keycode 28 = t T t
keycode 29 = y Y y
keycode 30 = u U u
keycode 31 = i I i
keycode 32 = o O o
keycode 33 = p P p
keycode 34 = dead_grave dead_circumflex bracketleft
keycode 35 = plus_asterisk bracketright
keycode 36 = Return
keycode 37 = Control_L
keycode 38 = a A a
keycode 39 = s S s
keycode 40 = d D d
keycode 41 = f F f
keycode 42 = g G g
keycode 43 = h H h
keycode 44 = j J j
keycode 45 = k K k
keycode 46 = l L l
keycode 47 = ntilde Ntilde
keycode 48 = dead_acute dead_diaeresis braceleft
keycode 49 = masculine ordfeminine backslash
keycode 50 = Shift_L
keycode 51 = ccedilla Ccedilla braceright
keycode 52 = z Z z
keycode 53 = x X x
keycode 54 = c C c
keycode 55 = v V v
keycode 56 = b B b
keycode 57 = n N n
keycode 58 = m M m
keycode 59 = comma semicolon
```

```

keycode 60 = period colon
keycode 61 = minus underscore
keycode 62 = Shift_R
keycode 63 = KP_Multiply
keycode 64 = Alt_L Meta_L
keycode 65 = space
keycode 66 = Caps_Lock
keycode 67 = F1 F11
keycode 68 = F2 F12
keycode 69 = F3 F13
keycode 70 = F4 F14
keycode 71 = F5 F15
keycode 72 = F6 F16
keycode 73 = F7 F17
keycode 74 = F8 F18
keycode 75 = F9 F19
keycode 76 = F10 F20
keycode 77 = Num_Lock
keycode 78 = Multi_key
keycode 79 = Home KP_7
keycode 80 = Up KP_8
keycode 81 = Prior KP_9
keycode 82 = KP_Subtract
keycode 83 = Left KP_4
keycode 84 = NoSymbol KP_5
keycode 85 = Right KP_6
keycode 86 = KP_Add
keycode 87 = End_KP_1
keycode 88 = Down_KP_2
keycode 89 = Next_KP_3
keycode 90 = Insert_KP_0
keycode 91 = Delete_KP_Decimal
keycode 94 = less greater
keycode 95 = F11
keycode 96 = F12
keycode 104 = KP_Enter
keycode 105 = Control_R
keycode 106 = KP_Divide
keycode 108 = Mode_switch
keycode 109 = Break
keycode 110 = Home
keycode 111 = Up
keycode 112 = Prior
keycode 113 = Left
keycode 114 = Right
keycode 115 = End
keycode 116 = Down
keycode 117 = Next
keycode 118 = Insert
keycode 119 = Delete
keycode 134 =
# Conseguido en R34.LINUX de FidoNet.

```

Otra cosa que hay que hacer es configurar bien los "exports" Eso se hace de una forma MUY cómoda para TODOS los usuarios del sistema añadiendo las líneas siguientes al /etc/profile o a cada usuario en su ~/.bash_profile

```
export LANG=es
```

6.4 Servidores de red

La mayoría de los servicios que se ofrecen en las redes Unix se ofrecen a través de programas individuales llamados *servidores*, *demonios* o *daemons*. Para que un servidor pueda funcionar se le debe asignar un número de puerto donde atenderá las peticiones que se le realicen, y necesita ser arrancado cuando el sistema arranque o bien cuando sea necesario. En este capítulo veremos la forma en que se configuran y trabajan los programas servidores en Linux.

6.4.1 El fichero /etc/services

El fichero */etc/services* es en realidad una base de datos relacional. Cada línea de ese fichero consiste en un nombre del servicio, un número de puerto para la conexión (donde el programa residirá), un nombre de protocolo y una lista de alias para ese programa.

Este fichero se utiliza tanto por los clientes como por los servidores. Algunos de los servicios que se encuentran en él no se suelen usar más, pero permanecen ahí por compatibilidad y para prevenir que otro programa ocupe su puerto cuando aún se puede estar utilizando en otros sitios de Internet. Este es un ejemplo (reducido) de la forma de un fichero */etc/services*:

```

# /etc/services
#
telnet 23 /tcp
smtp 25 /tcp mail
time udp /udp timeserver
...

```

Los servidores Unix/Linux establecen sus números de puerto mirando en este fichero la entrada correspondiente en cada servicio. Todos los servicios que se listan en el fichero son los considerados estándar, éste es, no es necesario que un servidor esté dado de alta en el fichero para correr como demonio.

6.4.2 Puertos reservados

A los puertos que ocupan los números 0 a 1023 también se les suele llamar *puertos reservados*. En Unix estos puertos están restringidos al superusuario. Esto quiere decir que un programa debe correr como root para utilizar cualquiera de estos puertos reservados (pero cualquier usuario puede conectarse como cliente a estos puertos).

La idea de tener estos puertos es para evitar que un usuario normal pueda escribir programas que simulen en puertos destinados a otros programas servidores y así recabar información privilegiada (como escribir un falso demonio de telnet que escuche en el puerto 23 las contraseñas). Hay que hacer notar que este concepto de puertos reservados no es un estándar de Internet, pero la casi totalidad de fabricantes de sistemas Unix adoptan este método como solución al posible problema.

6.4.3 El programa /etc/inetd

En las primeras versiones de Unix (BSD 4.2), para hacer funcionar un servicio de red se ejecutaban programas diferentes que atendían a cada uno. Al crecer el número de servicios que se necesitaban, se optó por una mejor idea, se empezó a utilizar un sólo programa llamado */etc/inetd* (El daemon de Internet). Este programa escuchaba en varios puertos a la vez y ejecutaba los servidores que se necesitaran en el momento en que se recibía la petición de conexión.

Inetd se ejecuta cuando la máquina arranca, formando parte del proceso de arranque mismo. Cuando empieza su ejecución mira lo que pone en el fichero */etc/inetd.conf* para determinar qué servicios de red debe controlar.

Para más información sobre como está estructurado este fichero es conveniente leer la página del manual correspondiente con *man inetd*.

6.2 Telnet

Una de las cualidades más potentes que ofrecen los sistemas operativos Unix es la posibilidad de tener una “*presencia remota*”, en una máquina cualquiera, ésto es, podemos ponernos a trabajar a través de nuestro ordenador en cualquier máquina en la que poseamos una cuenta y se halle ejecutando un *demonio* de telnet. Este demonio, se encuentra en el puerto 23 del servidor y gracias a él podemos conectarnos a esa máquina. Toda esta interacción sucederá en modo texto, ya que para interactuar a nivel gráfico deberíamos “*exportar un display*”, que es algo que se tratará más adelante en el curso.

La sintaxis para conectarnos a una máquina es muy simple, basta con hacer “*telnet jmáquina¿*”, y después de eso se nos pedirá el login y el password que tengamos asignados:

```
Acrobat:/ # telnet Acrobat
Trying 10.12.11.36...
Connected to Acrobat.uax.es.
Escape character is '^[]{}'.
```

```
Welcome to S.u.S.E. Linux 5.3 (i386)
-
Kernel 2.0.35 (ttyp3).
```

```
Acrobat login: temp
Password:
```

```
Last login: Fri Feb 26 17:42:30 on ttyp0 from 10.12.69.69.
No mail.
temp@Acrobat:/home/temp >
```

Como se puede ver hemos entrado en la cuenta del usuario *temp* con su password (que no aparece). Observamos como el sistema nos informa de cuando fue nuestro último acceso y desde donde, esto es muy útil para comprobar si alguien accedió a nuestra cuenta sin que nosotros lo supiéramos. Una vez que hemos entrado, podemos actuar con nuestra cuenta de igual manera que si estuviéramos en el servidor.

Existen unos comandos alternativos a telnet, son lo conocidos como *comandos-r*, entre ellos figuran *rsh*, *rlogin*, *rexec*, *etcm*. Aquí no los detallaremos por considerarlos desfasados y bastante inseguros (telnet también es bastante inseguro pero lo explicamos por su gran importancia en las comunicaciones). En su lugar recomendamos la instalación y uso de SSH (Secure Shell).

6.3 FTP

El FTP² es un comando que se basa en el protocolo con el mismo nombre y que sirve para copiar ficheros de una máquina a otra. Es un protocolo muy usado actualmente, y de hecho es de los más usados en Internet.

Su funcionamiento básico no reviste más dificultad que teclear *ftp jmáquina¿*. A continuación introduciremos nuestro login y nuestro password para conectar. Hay una modalidad que se llama **ftp anónimo** que consiste en acceder al servidor de ficheros sin necesidad de tener cuenta en él. Obviamente ésto debe ser permitido por el administrador del sistema. El motivo de que exista este modo de acceder es que hay muchas veces en las que alguien quiere permitir que muchas personas accedan a sus ficheros, y no puede ir creando cuentas para todo el mundo. Para utilizar este servicio (si está permitido), se debe introducir como nombre de usuario “*ftp*” o “*anonymous*” (Dependiendo de la versión), y como password tendremos que introducir nuestra dirección de e-mail.

Una vez dentro, podemos listar ficheros con “*dir*” o “*ls*”, movernos por los directorios con “*cd*”, y para bajarnos fichero emplearemos las órdenes “*get*” o “*mget*” (esta última permite comodines y bajarse varios ficheros a la vez). También podremos hacer uploads de ficheros con “*put*” o “*mput*”. Además, deberemos según el tipo de archivo poner el proceso en modo binario (con “*bin*”), o en modo ascii (con “*asc*”).

El FTP tiene muchas más opciones, para verlas todas es conveniente utilizar el *help* dentro del propio programa o también ver el manual (“*man ftp*”).

²File Transfer Protocol (Protocolo para la Transferencia de Ficheros)

```
export LC_ALL=es_ES.ISO-8859-1
```

Antes hablaba del XF86Config... Revisa el fichero a ver si tienes algo PARECIDO a esto:

```
Section "Keyboard"
    Protocol      "Standard"
    AutoRepeat    500 5
    LeftAlt       Meta
    RightAlt      Meta
    ScrollLock    Compose
    RightCtl      Control
    XkbKeycodes   "xfree86"
    XkbTypes       "default"
    XkbCompat     "default"
    XkbSymbols    "us(pc101)"
    XkbGeometry   "pc"
    XkbRules      "xfree86"
    XkbModel      "pc101"
    XkbLayout     "es"
EndSection
```

Con esto configuras el servidor para nuestro idioma, el teclado en las X, ...

Bueno... con todo esto creo que podrás poner las X en castellano... El tema de la “castellanización” del sistema LiNux en modo texto se sale del tema pero tiene que ver con un fichero que se llama “.inputrc”;-)

3.2.4 Apariencia

Ahora... hay que instalar el “gestor de ventanas”(WM)... más tarde entraremos en qué es eso... pero échale un vistazo otra vez al “xinitrc”:

```
if [ -x /usr/X11R6/bin/fvwm2 ]; then
    exec fvwm2
fi
if [ -x /usr/X11R6/bin/fvwm ]; then
    exec fvwm
fi
exec twm
```

Aquí se ve que si tienes el “fvwm2” lo arrancará y si no probará con “fvwm” para hacer lo mismo y solo en el caso de que no lo encuentre arrancará “twm”. Evidentemente “fvwm2”, “fvwm” y “twm” son algunos de los MUCHOS gestores de ventanas que hay... así que eres libre de modificar lo que quieras y usar cualquier “window manager”;-)

Creo que ya podemos tener las X funcionando...

3.2.5 Arrancando las X

Pues eso, una vez que ya está todo configurado... pulsa “startx” y entra en las X!!! :-))” Si hay algún problema revisa la documentación, lee los mensajes de error, pregunta a alguien, ... pégate con el SO, que es así como se aprende!! :-))” Para cambiar la profundidad de colores 8, 16, 24, ... bits deberas arrancar las X con “startx -bpp n” donde “n” es uno de los números anteriores y antes deberás haber configurado los modos en el XF86Config.

3.3 Modelo Cliente/Servidor

3.3.1 Introducción

Los sistemas UN*X son sistemas orientados a redes, multiusuario, multitarea, ... así que la orientación de los programas que corren en estos sistemas (LiNux es uno de ellos) es diferente a la de los programas que ejecutan en otros Sistemas Operativos (SSOO) monousuario, como W95 y monotarea como MSDOS.

La orientación a la que me refiero es el sistema cliente/servidor. Esto quiere decir que tendremos dos programas corriendo en vez de uno, con la particularidad de que podremos estar ejecutando uno de ellos (servidor) en una máquina estando nosotros utilizando el cliente otra que esté físicamente separada y cargando de trabajo al servidor mientras que la máquina delante de la cual estamos sea más sencilla y no sufre toda la carga de proceso.

Esto no quiere decir que no podamos tener el cliente y el servidor ejecutando en la misma máquina. Eso es lo que habitualmente haremos cada uno en nuestra casa cuando queramos tener nuestro ordenador en modo de ventanas.

Forma parte de la filosofía de los sistemas LINUX (UN*X en general) el funcionamiento de diferentes partes que cooperan de forma individual para conseguir un servicio integrado y completo.

3.3.2 Servidor de X (XFree86)

Básicamente está ya comentado todo lo que quería comentar de este apartado. Añadir también que para instalar las "X" en nuestra máquina deberemos instalar un servidor que funcione con nuestra tarjeta gráfica. Los servidores son todos iguales con la excepción de que están optimizados para uno u otro "chipset".

Así tendremos servidores para 3Dlabs, Rendition, Weitek, Cyrix MediaGX, Matrox, SiS, Trident, NeoMagic, Mach64, S3, También están los servidores para tarjetas "genéricas" como VGA, SVGA que no están optimizados para una tarjeta especial, así ganan en compatibilidad aunque puedan perder algo en eficiencia (que en la práctica no se nota). :-)

3.3.3 Clientes de X

El cliente es la parte que usa el usuario. Es un programa que se "conecta" con el servidor y que sirve para interactuar con el usuario. Una de las partes más importantes de este cliente es el "Gestor de Ventanas" que es un programa que se encarga de lo que su nombre indica. :-) Puede ser que veas escrito en algún lado "Window Manager" - "WM" ... es la traducción al inglés de gestor de ventanas.

3.3.4 Gestores de ventanas (window managers-WM)

Como decía antes es el programa que se encarga de ordenar, colocar, abrir, cerrar, ... GESTIONAR todas las ventanas de las aplicaciones que estemos ejecutando en un sistema de ventanas (X). Una de las múltiples ventajas de esto es que podemos cambiar este WM sin tener que tocar más que esto y tendremos, aparentemente, unas X distintas. Lo que hace esto es permitir cambiar la apariencia, funcionalidad según nuestras necesidades pudiendo elegir entre MUCHAS opciones diferentes. Recomiendo una visita a <http://www.PLiG.org/xwinman> donde podremos encontrar información detallada de muchos WM existentes así como enlaces a sus páginas de donde podremos bajárnoslos de forma totalmente gratuita.

Afterstep, fvwm, fvwm95, gnome, enlightenment, twm, ctwm, windowmaker, ... KDE.

No son más que varios ejemplos de gestores de ventanas entre los existentes. Entre ellos, por sus características particulares destacaré los siguientes:

- "twm". Este WM es uno de los más simples que existen y tiene la particularidad de la poquísima carga que representa para nuestra máquina. Eso sí, la poca carga que sufrirá quiere decir, evidentemente, que los servicios que ofrece son más limitados.
- "fvwm95". Ese "95" no es más que una pista de lo que trata de parecer este gestor de ventanas. Con él tendremos en nuestro ordenador una apariencia MUY similar a la que "disfrutábamos" con el "ventanucos 95 de MS".... Pero eso sí... SOLO LA APARIENCIA, eso no quiere decir que si instalamos el fvwm95 tengamos que soportar también los continuos cuelgues del Sistema Operativo (SO) o que aparezcan en nuestra pantalla los temidos "Errores de Protección general" vé tute a saber que errores del "kernel.386" NO!!! tranquilos X'DDDD

El que más me gusta: KDE

Aquí llegamos a un apartado especial en el tema de las ventanas... las KDE.

El proyecto KDE trabaja en construir lo que se llama un "Entorno de Escritorio" e incluye, entre otras cosas el gestor de ventanas, un gestor de ficheros, ... KDE no es solo un gestor de ventanas. El gestor de ventanas de KDE es Kwm.

KDE ofrece un sistema de aplicaciones integradas bajo una interfaz de usuario "estándar" lo que hace que todas las aplicaciones tengan un estilo similar y sea fácil de usar. Es un acercamiento al usuario doméstico. Las KDE han sido

```
Acrobat:/ # finger jacevedo@Acrobat
[Acrobat.uax.es]
```

```
Welcome to Linux version 2.0.35 at Acrobat.uax.es !
```

```
10:54am up 1 day, 2:26, 3 users, load average: 0.00, 0.08, 0.05
```

```
Login: jacevedo Name: Juan J. Acevedo
Directory: /home/jacevedo Shell: /bin/bash
Last login Fri Feb 26 15:09 (CEST) on tty0 from 10.10.200.221
No mail.
No Plan.
```

```
Acrobat:/ #
```

Vemos que se nos ofrecen distintos datos de usuario *jacevedo* en la máquina *Acrobat*, tales como la última vez que entró, su Nombre, el shell que utiliza, si tiene correo. Cada usuario puede escribir un fichero de nombre *plan* en su directorio de usuario, de ese modo cada vez que alguien le haga un *finger* aparecerá lo que haya escrito allí.

Por último, se puede leer la sintaxis de este comando *jacevedo@Acrobat* de la siguiente manera: "*usuario jacevedo en la máquina Acrobat*".

6.1.4 Netstat

El comando *netstat* se utiliza para varias cosas, pero la más habitual de ellas es ver las conexiones que nuestra máquina recibe del exterior o ella misma hace fuera. Su sintaxis es algo más compleja, aquí detallaremos la más simple. Si el lector quisiera más información acerca de este comando, puede ejecutar *man netstat*. Si por ejemplo, quisiéramos una simple lista de las conexiones abiertas:

```
Acrobat:/ # netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 Acrobat.uax.es:28913 Acrobat.uax.es:6667 ESTABLISHED
tcp 0 0 Acrobat.uax.es:6667 Acrobat.uax.es:28913 ESTABLISHED
tcp 0 0 Acrobat.uax.es:28915 Acrobat.uax.es:6667 ESTABLISHED
tcp 0 0 Acrobat.uax.es:6667 Acrobat.uax.es:28915 ESTABLISHED
tcp 0 0 Acrobat.uax.es:6667 10.12.33.52:1036 ESTABLISHED
tcp 0 0 Acrobat.uax.es:telnet 10.10.200.221:8312 ESTABLISHED
tcp 0 0 Acrobat.uax.es:6667 10.12.23.62:1191 ESTABLISHED
tcp 0 0 Acrobat.uax.es:6667 10.12.32.6:1165 ESTABLISHED
tcp 0 0 Acrobat.uax.es:6667 10.12.31.16:1192 ESTABLISHED
udp 0 0 Acrobat.uax.:netbios-ns *: *
udp 0 0 Acrobat.uax.:netbios-dgm *: *
udp 0 0 localhost:9090 *: *
```

Lo que aparece arriba es una parte de la salida que produciría el comando *netstat*, se pueden ver las conexiones que existen activas, y se indican entre otras cosas el protocolo, la máquina y puerto origen, la máquina y el puerto destino, y el estado actual de la conexión.

Además, *netstat*, se puede utilizar para conseguir una lista de los interfaces de red que se encuentran activos en la máquina, esto se hace con la opción *-i*, y se puede mostrar la tabla de enrutamiento con la opción *-r*:

```
Acrobat:/ # netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
10.12.0.0 * 255.255.0.0 U 1500 0 0 eth0
loopback * 255.0.0.0 U 3584 0 0 lo
default 10.12.99.8 0.0.0.0 UG 1500 0 0 eth0
Acrobat:/ #
```


mandado, los que se han devuelto, su porcentaje, y los valores mínimo, medio y máximo de respuesta. Si por ejemplo, esa interfaz no hubiera estado activa, hubiéramos tenido la siguiente respuesta (después de haber pulsado *Control-C*):

```
Acrobat:/ # ping apolo
PING apolo (10.12.99.8): 56 data bytes

---apolo.uax.es ping statistics ---
1 packets transmitted, 0 packets received, 100\% packet loss
Acrobat:/ #
```

Lo que se está haciendo en realidad es mandar paquetes del tipo “*echo request*”, y para los que se devuelven son del tipo “*echo reply*”. Para más información sobre esto se puede consultar el RFC 792 y el RFC 777.

6.1.2 Traceroute

Este es un comando que se utiliza para comprobar el camino que siguen nuestros paquetes para llegar a la máquina (interfaz destino. Su funcionamiento es simple (aunque existen muchas opciones). La sintaxis es *traceroute jm máquinaζ*. De este modo, si supongamos que quiero ver desde *Acrobat* por donde pasan los paquetes para llegar a 193.146.25.2 (esta salida es ficticia):

```
Acrobat:/# traceroute 193.146.25.2
traceroute to 193.146.25.2 (193.146.25.2),
30 hops max, 40 byte packets

 1 10.12.99.8 (10.12.99.8) 0.469 ms 0.319 ms 0.339 ms
 2 10.12.76.27 (10.12.76.27) 0.539 ms 0.378 ms 0.139 ms
 3 193.146.25.2 (193.146.25.2) 0.234 ms 6.201 ms 0.932 ms
Acrobat:/ #
```

La salida se puede ver como la lista de máquinas por las que han pasado los paquetes ICMP que hemos enviado y una serie de tiempos que nos indican algunas estadísticas de la transmisión.

Traceroute es una utilidad bastante práctica si siendo administradores de una red queremos ver por donde van nuestros paquetes, y de esa forma corregir cualquier fallo en el enrutado que pueda haber.

6.1.3 Finger

El comando *finger* se utiliza para recabar información acerca de quienes se encuentran en una máquina determinada, o saber más acerca de un usuario en esa máquina. Su sintaxis puede variar, por ejemplo si quisiéramos saber quien está en *Acrobat* desde *apolo*:

```
Acrobat:/ # finger @Acrobat
[Acrobat.uax.es]
```

```
Welcome to Linux version 2.0.35 at Acrobat.uax.es !
```

```
10:45am up 1 day, 2:17, 3 users, load average: 0.10, 0.06, 0.01
```

Login	Name	Tty	Idle	Login Time	Office	Office Phone
root	root	1	1d	Mar 2 08:29		
root	root	p0	10:39	Mar 2 13:50		
root	root	p1		Mar 2 13:55		

```
Acrobat:/ #
```

Vemos que en *Acrobat* se encuentra ahora *root* (el superusuario), con tres sesiones diferentes. Además se indican datos tales como la carga del sistema (para hacernos una idea de como está siendo utilizada la CPU de la máquina). Además, si estuviera la información aparecerían algunos datos como números de contacto, etc ... Si ahora por ejemplo quisiéramos información acerca del usuario *jacevedo*, esto es lo que obtendríamos:

polémicas dentro del mundo del LiNux por varias razones, entre ellas que están basadas en unas librerías (las QT) que hasta hace bien poco no seguían la licencia GPL, así que las KDE tenían ciertos “problemas” con el tema de la licencia. Otro “problema” que ha podido tener KDE es que sea “demasiado guiindosero”, que sea demasiado orientado a gráficos, ... tal y como lo es W95 de M\$. Eso no quiere decir que tenga problemas, porque como todos sabemos el problema del W95 no es el entorno de ventanas, sino el SO sobre el que corre (que dicho sea de paso no deja de ser un sistema MSDOS).

Pero bueno, esto no es más que un ejemplo de las diferentes opiniones que acerca de las KDE se escuchan (y se leen). :-)

Las KDE se pueden obtener bien de su servidor de Internet (Inet a partir de ahora) en <http://www.kde.org> (o en uno de sus múltiples mirrors) o en muchos de los CDs que vienen con las revistas.

En la actualidad está disponible la versión 1.2 estable. Habiéndose distribuido otras versiones “beta” anteriormente que aunque no eran estables oficialmente... funcionaban MUY bien :-))

Está previsto el lanzamiento de las KDE-2, pero en estos momentos se encuentran en fase de desarrollo. KDE-2 (KRASH) supondrá un gran avance en el acercamiento al usuario ya que los equipos de programación han reescrito muchas partes del código, optimizando rendimientos y aumentando funcionalidades.

El entorno KDE consta de MUCHOS paquetes con software para muchas utilidades y es MUY posible que NO NECESITEMOS TODOS. Aun así hay algunos que son básicos para el funcionamiento global. Estos son: kbase, klibs, ksupp y por supuesto tener instalada la última versión de las librerías QT que podrás encontrar en <http://www.troll.no>

A parte de estos paquetes disponemos también para instalar otros como:

kadmin Utilidades de administración.

kgraphics Pues eso, gráficos.

kmultimedia Paquete para aplicaciones multimedia.

knetworks Utilidades relacionadas con redes (correo, ...)

kgames Juegos, ...

kutils Utilidades varias, ...

... Y otros paquetes.

Entre las aplicaciones que podemos encontrar en las KDE me gustaría resaltar algunas que por su utilidad me parece importante hacer conocer.

Entre esas aplicaciones están: KMail, Kppp, Knews, ...

KMail es un programa para leer el correo. Si conoces o eres (o has sido hasta ahora) usuario de W95 de M\$ es muy posible que conozcas programas como el “Eudora”, “Pegasus”, ... o las partes relacionadas con el correo electrónico de “Outlook”, “Netscape” “Explorer”...

Kppp es un programa que te ayudará a utilizar el demonio “pppd” que es el que tendrás que utilizar para conectarte a Inet a través de una línea de teléfono, ya sea a través de Infovía o de Retevisión o de cualquier nodo local de cualquier otro Proveedor de Servicios de Internet (ISP).

KRN es otro lector de noticias integrado en el paquete “KDENetworks”. Es básicamente igual al KNews, pero este está en castellano (como la mayoría de las aplicaciones de KDE) y es algo más “bonito” y fácil de usar que KNews.

3.4 Aplicaciones para las X

Yo vengo del guiindous... ¿Cómo puedo migrar a LiNux? ¿Qué puedo hacer con LiNux? Seamos sinceros, LiNux aun no está al mismo nivel de “estandarización” de los productos M\$, así que existen programas que corren en sistemas M\$ que no existen en plataformas LiNux...

Eso sí, si puedes vivir sin los juegos, puedes vivir sin M\$. Eso es lo único que se puede llegar a echar en falta si funcionas EXCLUSIVAMENTE con LiNux... los juegos (aunque existen emuladores) :-) El resto de actividades, funcionalidades, utilidades, aplicaciones que pudieras necesitar tener en tu PC, las encontrarás en LiNux.

3.4.1 Office

¿Quién no conoce el MS Office en cualquiera de sus múltiples versiones? Pues los usuarios de LINUX también disfrutamos de nuestros procesadores de texto, hojas de cálculo, ... pero a un precio MUCHISIMO más bajo... y por supuesto, sin perder funcionalidad.

StarOffice.

Es una "suite" al estilo del MS-Office... Está ya en su versión 5.1. La URL donde informarnos de este programa es: "www.stardivision.com". SUN (<http://www.sun.com>) compró a mediados de 1999 la compañía StarDivision, así que también es posible obtener el programa desde SUN. Allí podremos registrarnos y bajarnos el programa de forma totalmente gratuita. Tengo que decir que el programa es MUY extenso, por lo que la carga que soporta la máquina cuando ejecutas SO 5.1 es grande. Eso sí, con este programa se acaban muchas de las "excusas" que de los "winusers" (usuarios de windows) se escuchan... - "Todo el mundo funciona con MS... ¿Que haces si te envían un fichero *.doc de WORD? ... SO 5.1 es capaz de abrir, escribir,... ficheros de MS Office!!!! :-)" Entre otros MUCHOS formatos claro... :-) StarOffice cuenta con un procesador de textos, hoja de calculo, programa de correo,... todo ello compatible con los "formatos propietarios" de MS, así mismo cuenta con hoja de cálculo, gráficos, programa de correo, explorador de ficheros, ...

WordPerfect (Corel).

Pues sí, WordPerfect también está disponible en LINUX, por lo menos desde su versión 6. Eso sí, la versión 6 era de pago :- (Claro que... con el nuevo "boom" de LINUX hay cada vez mas empresas que están ofreciendo su apoyo a LINUX y "Corel", actual propietaria de WordPerfect no se ha querido quedar atrás y la versión 8 de WordPerfect está disponible de forma gratuita en la red para los usuarios de LINUX en: <http://linux.corel.com>. También es posible conseguir WP8 en los CDs que distribuyen algunas revistas lo que nos hará ahorrarnos la factura telefónica que nos causaría el tener que bajarnos los megas que ocupa. La instalación es MUY sencilla y luego solo requiere un registro en línea (Inet) para poder usar esta versión de WP8 en nuestra máquina de forma totalmente gratuita y sin copias "raras". Una vez instalado el WP8 contaremos con lo mismo que se ofrece en el resto de procesadores de texto del mercado... fuentes de letras, paginación, corrección ortográfica, gráficos, posibilidad de guardar en distintos formatos (incluidos los usados por Office97)...

Aplixware.

No puedo dar mi opinión personal sobre este software pues es software comercial y no he tenido oportunidad de conocerlo aun.

KOffice.

Es la suite de KDE... Este paquete de programas está en fase de desarrollo, así que aun queda mucho por hacer, pero conociendo a la gente que trabaja en KDE acabará haciéndose un hueco entre las suites de oficina para LINUX. Trata de integrar toda la suite en el entorno KDE y consta de procesador de textos (kword), hoja de cálculo (Kspread), presentaciones (kpresenter), gráficos (killustrator), ... Por ahora es inestable y personalmente NO lo recomiendo a no ser que sea para curiosear y ver como va desarrollándose (pero estoy escribiendo esto en Marzo del 2000, veamos que pasa de aquí a unos meses ¿eh?)

NExS.

Esto es una hoja de cálculo, bastante parecida a lo que puede ser EXCEL para que os hagais una idea. Este programa NO es de libre distribución, así que tened cuidado con lo que instaláis y de donde lo sacáis, porque una de las "gracias" de LINUX es poder tener un PC con TODO el software instalado de forma legal :-)" Se acabaron las "WaReZ RuLeZ". :-)

3.4.2 Emuladores

Antes de nada... ¿Qué es un emulador? Pues básicamente, un emulador es un programa que te permitirá ejecutar otros programas de unos SSOO como son MS-DOS o MacOS sobre otras plataformas. En LINUX disponemos de emuladores para distintos SSOO. Este apartado no es exclusivo del tema de las X, pero bueno... O:-)

Capítulo 6

Trabajando en red

Hasta ahora hemos introducido sólo unos conceptos teóricos fundamentales sobre las redes TCP/IP y la forma de configurar nuestro hardware para que se tenga acceso a la comunicación, Ahora, en este capítulo, aprenderemos a utilizar algunos de los comandos más útiles cuando se trabaja en red. Además, iremos introduciendo paulatinamente otros conceptos sobre redes.

6.1 Comandos útiles

A continuacúon, presentaremos algunos comandos que se emplean de forma muy habitual para trabajar en red. Estos comandos son típicos de cualquier sistema Unix, y aunque la salida que producen pueden variar de un sistema operativo a otro, su utilidad y modo de funcionamiento son esencialmente los mismos.

Aunque aquí se va a comentar cada comando de forma general, la mayoría dispone de muchas más opciones de funcionamiento, si al lector le interesase conocer más acerca de algún comando, debería acudir a las páginas del manual con: *man comando*.

6.1.1 Ping

Sin lugar a dudas en una red el comando que más se puede llegar a utilizar es *ping*. Este comando se utiliza para comprobar si una determinada interfaz de red, de nuestro ordenador o de otro, se encuentra activo. Su función más habitual es simplemente verificar si una máquina está encendida. Pongamos un ejemplo, si mi ordenador es *Acrobat* (10.12.11.36) y quiero ver si *apolo* (10.12.99.8) se encuentra activo (al menos esa interfaz), utilizaría *ping*, y su salida sería:

```
Acrobat:/ # ping apolo
PING apolo.uax.es (10.12.99.8): 56 data bytes
```

```
64 bytes from 10.12.99.8: icmp_seq=0 ttl=64 time=0.4 ms
64 bytes from 10.12.99.8: icmp_seq=1 ttl=64 time=0.3 ms
64 bytes from 10.12.99.8: icmp_seq=2 ttl=64 time=0.3 ms
```

```
--- apolo.uax.es ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.4 ms
Acrobat:/ #
```

Lo que hemos hecho es teclear el comando, y pulsar *Control-C* para parar la ejecución del comando (si no, se ejecutaría indefinidamente). Lo que se puede ver es que la interfaz 10.12.99.8 se encuentra activa. Cada una de las líneas que comienzan por "64 bytes from 10.12.99.8 ...", nos está mostrando que el paquete ICMP¹ que hemos mandado a *apolo*, nos ha sido devuelto. Al final de cada una de esas líneas se encuentra "*time=xx ms*", que nos indica el tiempo de respuesta que ha tenido ese paquete en milisegundos. Al final se nos muestra una estadística de todos los paquetes, los que se han

¹ICMP (Internet Control Message Protocol), Protocolo de Mensajes de Control.

Wine: Emulador de Windows 95.

Te permitirá ejecutar aplicaciones y programas de ese SSOO en nuestra máquina LINUX... Wine no está completamente desarrollado y aun no es capaz de ejecutar aplicaciones grandes como M\$WORD u otras del mismo estilo, así como los juegos que tanto se echan en falta en LINUX... De todas formas hay muchas aplicaciones que si se pueden ejecutar, así como juegos sencillos. Podrás encontrar el fichero de configuración, tras instalar el software en /etc/wine.conf Es un programa de libre distribución.

Wabi: "Emulador" de windows 3.1

He entrecomillado "Emulador" porque no sé si se le puede llamar así. Me explico, este programa te permite instalar Windows 3.1 en tu maquina LINUX, y cuando digo instalar es eso, INSTALAR. Te pedirá los discos originales de win3.1 como si lo estuvieras instalando en una máquina con MSDOS. De la misma forma que el wine, tampoco te permitirá ejecutar las aplicaciones más complejas de ese SO aunque debido a que las aplicaciones de w3.1 son más sencillas, te dará mas margen.

Dosemu

De acuerdo que el dosemu no es un emulador bajo "X"... así que hablare de "xdosemu" para no salirme de la temática del capítulo X'DDDD El dosemu, como habrás adivinado (espero) es un emulador de MSDOS. De los emuladores que hemos tratado hasta ahora es el que mayor grado de desarrollo ha alcanzado.

Te ejecutar gran número de programas de MSDOS... programas de correo, comunicaciones, algunos juegos, ... Puedes emular sonido, ... Vamos que da bastante más juego que el wine o el wabi :-)

Hay gente que no quiere prescindir de su programa de correo para FidoNet preferido y escribe/envia/recibe correo con LINUX a través de dosemu. Encontrarás el fichero de configuración en /etc/dosemu.conf, pero ten cuidado con lo que tocas porque puede resultar un "fallo de seguridad" si le das acceso al emulador cualquier persona o acceso a cualquier disco o partición dentro del dosemu... }:-) Este programa también es de libre distribución.

Executor

Este es el emulador más potente de otros SSOO de los que conozco dentro del LINUX. Executor emula BASTANTE BIEN el MacOS. Con este programa podrás ejecutar casi cualquier programa (si no cualquier) programa para "Mac"... :-)"

Yo he visto ejecutar Photoshop y Lemmings en una máquina LINUX corriendo Executor desde las KDE... la leshe!! :-)"

XMame

¿Conoces el mame? Bueno, pues este emulador de juegos de arcade también lo tienes disponible para LINUX así que con el podrás jugar a muchos juegos (si consigues las ROM's)

VMWare

El emulador. Vmware es un emulador de máquina virtual, me explico. Vmware emula por software un PC, con una BIOS y todo... eso quiere decir que arrancando LINUX, las X y dentro de las X, en una ventana, podrás arrancar un nuevo "PC" donde podrás a su vez instalar MSDOS, Win95/98, NT, 2000.... Vmware accede directamente al hardware, así que es posible acceder a todos los recursos de la máquina.

Es un programa comercial de pago, aun así es posible obtener una licencia evaluación de 30 días. En estos momentos acaba de salir la versión 2.0 y es totalmente funcional y muy estable... vamos que se te colgará el Windows en la máquina virtual antes que colgarse la máquina virtual. O:-)

3.4.3 Gráficos

En esta parte trataré de contaros que es lo que hay a nivel de tratamiento, visualización, ... de gráficos dentro de LINUX bajo las X.

GIMP

GIMP significa "GNU Image Manipulation Program" que para la gente a la que no le guste mucho el inglés, traduciré como "Programa GNU de Manipulación de Imágenes".

Este programa, que ya ha alcanzado su versión 1 (estable) está basado en el Photoshop, pero cuando digo que está basado, no estoy diciendo que no llegue a su nivel...

Personalmente no entiendo mucho de retoque fotográfico, ni de manipulación de imágenes y esas cosas, pero he oído maravillas de este programa. A primera vista, y para los no entendidos en la materia, el programa es MUY sencillo de usar y te permitirá actuar sobre ficheros de gráficos de múltiples formatos así como convertir un formato en otro.... También puede usarse como programa de dibujo, tiene las típicas opciones, herramientas, brochas, filtros, ...

XView

Este programa es un "visualizador"de gráficos de muchos formatos como los BMP, JPG, GIF, ... incluso ficheros PostScript para las X. El programa arranca tecleando "xv".

Gqview

También es un "viewer"como el XView, pero más sencillo.

3.4.4 Utilidades

Entre las utilidades que podemos encontrar para LINUX... que son básicamente las mismas que podemos encontrar para otros SSOO están:

Mathematica

El programa para hacer cálculos matemáticos, tanto numéricos como simbólicos. Es un programa MUY potente. Es un programa de pago :-)

Mat-Lab

: Creo que todos o casi todos conocemos este programa en sus versiones para MS Windows ¿no? Pues eso, también disponemos del programa en su versión para LINUX... Evidentemente, es un programa de pago. :(De todas formas existe el "ctave"que es un programa de libre distribución MUY similar a MatLab, aunque no es totalmente compatible.

XMMS

El antiguo X11Amp. ¿Conoces el formato de ficheros de sonido "MP3"? ¿Conoces del Windows el "Winamp"? Pues el XMMS es un programa MUY similar que te permite escuchar ficheros.mp3 con un front-end similar al del winamp. Permite generar listas de canciones, modificar los "skins"(la apariencia), repetición de canciones, escucharlas en orden aleatorio... :-)) Pásate por... www.xmms.org para saber más :-)

Real Player

Es un programa para aprovecharse del formato "RealAudio"que se está utilizando ahora en Inet. No lo he visto, pero se que existe O:-)

Acrobat Reader

¿Alguien no lo conoce? Por si las moscas comentaré que es un programa de "Adobe"que permite leer los ficheros con formato *.PDF. Bueno, pues Adobe también se acuerda de LINUX y nos brinda la posibilidad de disfrutar de su lector.

5.5 El protocolo AX25

El AX25 es un protocolo de comunicaciones que está pensado para utilizarlo con TNCs. Las TNCs son equipos parecidos a los módems que permiten una comunicación vía radio. Las principales diferencias entre una comunicación vía radio y otra por los cables telefónicos son:

La radio es half-duplex. Las comunicaciones pueden realizarse en ambos sentidos, pero sólo uno a la vez. Las colisiones provocan que ambos paquetes se pierdan. Además a diferencia de redes tipo Ethernet, la colisión no se puede detectar y por tanto, no se interrumpirá. La radio es un medio broadcast. Mientras que un módem es punto a punto, y sólo dos participan en la transmisión, en la radio puede haber cualquier número de personas. Esto nos obliga a utilizar direcciones, (en PaP no hacen falta), y que el protocolo tenga en consideración respetar unos tiempos muertos para que puedan transmitir todos.

AX25 es un protocolo fiable y orientado a conexión. Aunque existe también la posibilidad de enviar paquetes en modo no conectado. Una vez que se establece la conexión, AX25 nos asegura que los datos llegan correctamente, o que se abortará la conexión. Utiliza rechazo simple, por lo que si un paquete se pierde, el resto se tirará aunque lleguen bien.

Las direcciones que se usan en este nivel son los propios indicativos de los radioaficionados, ya que es un protocolo diseñado para ellos. (¿y por ellos?). También existen 16 identificadores, que son muy semejantes a los puertos en TCP. Aunque son pocos, la red de packet es muy lenta y no puede soportar un gran número de servicios. Además, a diferencia de TCP, un mismo identificador se puede utilizar para establecer varias conexiones, o bien con distintas personas, o bien con la misma persona pero a distintos identificadores.

Sin embargo, en lo que a Linux se refiere, la idea más atractiva del AX25 es poder usarlo como nivel de red de la pila de protocolos. Configurando esto, se puede usar IP metido en AX25, igual que lo harías en Ethernet, pudiendo establecer una red completa en IP, con enrutamientos, servicios, etc.

Esto funciona muy bien en teoría pero en la práctica no es muy rápido. Al encapsular los datos en TCP y luego IP, estamos ocupando unos 40 bytes. La MTU de AX25 es de 256 bytes, con lo que una buena parte del mensaje son cabeceras. Sin embargo, es posible construir TNCs más rápidas, con las que se podría obtener un mejor rendimiento y montar una red sin cables. Además con el enrutamiento y enmascaramiento de Linux, se podrían hacer cosas como: Compartir una sola conexión a internet para que entrasen varios equipos, de tal forma que pudieran tener su propia IP (si la conexión es en red local) o que usasen enmascaramiento (si la conexión es PPP a un proveedor de internet).

Las versiones actuales de Linux incorporan el soporte para AX25 ya en el kernel, con lo que sólo hay que compilarlo para que funcione, y por supuesto, configurar algunos ficheros.

El más importante es /etc/ax25/axports, donde configuramos el nombre del interface, la velocidad del puerto de serie, nuestro indicativo, etc.

También hay que usar algunos comandos para que nuestra TNC se convierta en un interface de red más

Un fichero clásico de configuración sería:

```
kissattach -i 44.0.0.1 -m 512 /dev/ttyS3 radio
```

Configuramos el interface como ax0, con la IP 44.0.0.1

```
kissparms -p radio -t 200 -s 0 -r 0
```

Le damos algunos parámetros, como el timeout, el tiempo de espera antes de transmitir, etc.

```
route add -net 44.0.0.0 netmask 255.255.255.0 ax0
```

Configuramos la tabla de enrutamiento, en caso de tener más de un interface de red.

Y con todo esto deberíamos ser capaces de hacer un ping a nuestra propia máquina, o a otra configurada de esta forma, y el interface de red eth0 debería estar en el ifconfig, el route, podríamos verlo con el tcpdump, etc.

Los módulos para tarjetas de red se encuentran en el directorio `"/lib/modules/2.0.xx/net"`. Se puede probar primero a cargar el módulo sin parámetros y ver si funciona (`"insmod ne"`). Si da algún mensaje de error, es necesario indicarle la dirección de entrada/salida y la IRQ. Si la tarjeta es de tipo PCI, esta información se puede encontrar en `"/proc/pci"` o en los mensajes que da la BIOS al arrancar el ordenador. Si es ISA, buscad en los manuales o en los `"jumpers"` de la propia tarjeta. (Un jumper es una conexión que se puede cambiar a mano, cambiando una piececita negra de sitio. Se usa en discos duros, placas, tarjetas de ampliación ... para configurar a mano ciertos parámetros hardware).

5.4.2 Configurar la red

Vamos a suponer un caso totalmente irreal, en el que nos encontramos en una residencia universitaria con conexión de red en todas las habitaciones. El C.P.D. (Centro de Proceso de Datos) nos ha dado un papelito con los datos de nuestra conexión:

- Dirección IP: 10.11.12.13
- Máscara de subred: 255.255.0.0
- Gateway: 10.10.99.8
- DNS: 10.10.99.9

Esto es todo lo que necesitamos para configurar la red.

En la mayoría de las distribuciones actuales, hay algún programa más o menos intuitivo para configurar la red (Yast en la Suse, linuxconf en la RedHat, netconfig en la Slackware ...). Dado el carácter abierto de este curso, indicaré cómo se debería hacer en un caso general.

La dirección IP es la que corresponde a nuestro ordenador, o más concretamente, al interfaz que comunica nuestro ordenador con la red interna de la residencia, o sea ... nuestra tarjeta. Al instalar una tarjeta ethernet, Linux le asigna un nombre de interfaz de red: `"eth0"`. Primero, asociamos la dirección IP que nos han dado, con el interfaz:

```
ifconfig eth0 10.11.12.13 netmask 255.255.0.0 up
```

El Gateway es la máquina que se va a encargar de enrutar todo el tráfico de datos que generemos y mandarlo a su destino por el camino más adecuado. Modificamos nuestra tabla de enrutamiento para que, por defecto, mande todo a la ethernet, con destino al gateway.

```
route add default eth0
route add default 10.10.99.8 eth0
```

El DNS se encargará de traducir los nombres de máquinas en direcciones IP, que son las que entiende la red. Añadimos la siguiente línea al fichero `"/etc/resolv/conf"`

```
nameserver 10.10.99.9
```

Cada vez que se reinicie la máquina, la asignación hecha con `"ifconfig"` y las modificaciones en la tabla de enrutamiento se pierden. Es por esto, que se hace necesario incluir los órdenes anteriores en algún script que se ejecute cada vez que Linux arranca.

5.4.3 Notas

Todo lo dicho en los apartados anteriores sobre la configuración de una red local es totalmente aplicable al caso de una red con Ethernet. La única diferencia está en el cambio de nomenclatura:

```
plip0 ---> eth0
```

Si lo que usamos para conectar los dos ordenadores es Ethernet, la restricción de trabajar únicamente en Linux o MsDOS desaparece. Con ethernet es posible conectar una máquina en Linux con otra en Windows 98 y usar todo el software de red que existe para este Sistema operativo. Además, instalando el paquete de `"Samba"` para Linux, se puede acceder a cualquier máquina que tenga activada la opción de `"compartir archivos e impresoras"`.

Para más información sobre configuración de redes lo mejor es que el lector se dirija a los textos del Proyecto LUCAS (Linux en Castellano) en <http://lucas.hispalinux.es>, y que busque la GARL (Guía de Administración de Redes en Linux), que es un excelente texto para todo lo relacionado con las redes en Linux, aunque quizás esté algo desfasado.

XTerm

Aunque estemos trabajando en modo X, alguna vez es posible que necesitemos teclear algo, ya sea para usar un práctico lector de email (pine) o para leer nuestro correo de FidoNet (feddi) o para editar algún fichero con el vi (o para los más `"cobardes"` con el joe), o arrancar algún programa, ... Bueno, es igual para qué, simplemente `"xterm"` brirá un terminal virtual en el que encontraremos una shell en la que algunos se sentirán más cómodos por la reaparición de las letras X'DDD

XEyes

Esto no es más que una `"chorradita"`, pero es muy típico de las Xwindow... tecllea `"xeyes"` y MIRA X'DD

XClock

Es otra de las aplicaciones `"típicas"` dentro del mundo UN*X. Simplemente es un reloj :-)

Bueno, me dejo otras muchas aplicaciones... pero es que si no necesitaríamos más espacio y tiempo del que dispongo. O:-)

3.4.5 Redes

Este es el plato fuerte de nuestro SO, LiNUNIX, las redes. Empezaré por los programas para Internet...

Navegadores

Aquí hay MUCHO donde elegir...

Netscape

Evidentemente, el mismo que el que hemos visto para otros SSOO. Tenemos las distintas versiones, Communicator, Navigator, ... con los mismos problemas, ventajas e inconvenientes que sus iguales en otros SSOO.

Si instalamos en nuestra máquina el NS Communicator, dispondremos, además del típico navegador para la web, un lector de correo y un lector de noticias que (y para los cómodos os diré que usar un programa de estos para el correo y las news os ahorrará el tener que configurar el servidor de correo y news en vuestro sistema y que, aunque no es la solución más elegante, funciona bien), pero tendremos que soportar un mayor tiempo de espera mientras carga y una mayor carga de trabajo en nuestra máquina.

Por otra parte, con el Navigator, sólomente dispondremos de un navegador para la www, pero podremos utilizar los recursos de nuestra máquina de forma más `"eficiente"`.

Actualmente está disponible en `"ftp.netscape.com"` la última versión 4.7, tanto del Communicator, como del Navigator.

Arena

Este es otro navegador para la www. Es un navegador MUCHO más sencillo que el Netscape, pero también más rápido en arrancar y con menos carga para nuestra máquina.

Este navegador NO soporta todas las facilidades que empiezan a hacerse hueco en la web, así que aunque es interesante conocerlo, no lo recomiendo como navegador por defecto a no ser que las páginas que suelas visitar no incluyan `"frames"` y demás... :-

Hoy por hoy el IExplorer no está disponible para ser utilizado en LiNUNIX por razones obvias :-)

[... Un par de meses después, en la recisión de este texto...] He leído en algún sitio que MS ha portado su explorador a UN*X... pero no he tenido el `"placer"` de verlo aun }:-)

ICQ

Si, también lo tenemos disponible los usuarios de LiNUNIX!!! Es el ya famoso programa que te avisará cuando alguno de tus amigos, conocidos o como los quieras llamar, que tengas en una lista estén conectados Internet, permitiéndote enviar mensajes, ficheros, abrir sesiones de `"chat"`...

Esta versión de ICQ NO es exclusiva para LiNUNIX... Está programado en JAVA, por lo que tendrás que tener instalado el intérprete de JAVA... Yo tengo instalado `"jdk 1.1.3"` y me funciona PERFECTAMENTE... Eso sí, con la 1.1.6 no

funciona tan bien... no sé por que aunque si se puede ejecutar con la 1.1.6v5 :-? posiblemente se trate de algún "bug"de esa versión, porque con versiones posteriores vuelve a funcionar perfectamente.

IRC

Internet Relay Chat ¿Cómo iba a faltar entre las aplicaciones de redes para las X los clientes para IRC?

Tenemos MUCHOS distintos entre los clientes gráficos como KIRC, ksirc... Cada uno con sus ventajas e inconvenientes... te recomiendo que los pruebes y elijas el que más te guste... o que uses un cliente en modo texto a través de una xterm, que puede resultar también una buena solución.

XFMail

Es un lector de correo MUY popular. Este lector está en una fase avanzada de su desarrollo lo que le permite ser unos de los lectores de correo para las X más estables que conozco, con funcionalidades MUY diversas... incluso soporte para PGP :-)" Necesita que tengamos las librerías "xform";-)

Kscd

El reproductor de CD's de audio. Con el podrás escuchar los CD's de música en tu ordenador.

3.5 Para los radioaficionados

Twpsk

Este es un programa para practicar PSK31, es una aplicación que funciona en entorno X y es posible hacer ajustes visuales para sintonizar las señales entre el ruido de una forma tan cómoda como lo es en los programas de W9x.

DHLog

Es una aplicación de libro de guardia muy configurable y fácil de usar.

Qsstv

Este programa de SSTV, en su versión 3 incorpora bastantes funcionalidades. La principal pega que le veo, personalmente a este programa es la carga que hacer que la máquina tenga que soportar, perdiéndose, de vez en cuando, el sincronismo de la imagen recibida.

3.6 Trucos

Propiamente dicho, lo que aquí encontrarás no son trucos sino cosillas que no suelen aparecer por ahí, pero que están y agilizan mucho algunas cosas.

- Pulsando "CTRL+ALT+BORRAR"(sin las comillas, claro) ;-) matará el proceso del servidor de X, lo que causará tu retorno inmediato a la consola desde la que lanzaste "startx"para arrancar las X.
- Con "CTRL+ALT+Fx"conseguirás llegar al "modo texto". La x de "Fx"es el número del terminal virtual (tty) al que quieres ir... usualmente este número suele estar entre 1 y 6. Para volver a las X, encontrándotelas en el mismo estado en el que las dejaste, pulsa "ALT+F7"ya que es el el terminal 7 en el que se muestra el "display"de las X.
- "ALT+F1"Así (al menos con KDE) abrirás el menú desplegable de la barra inicio.
- Pulsando "ALT+F2"podrás introducir un comando. Esto es muy útil para arrancar algún programa para el que no tienes icono sin necesidad de abrir una XTerm.
- "ALT+F4"Cierra la ventana que estás usando en este momento.

Jace tiene que coger todos los paquetes que le llegan de *peque* con dirección a Internet, cambiarles la dirección de origen para que parezca que son suyos y enviarlos a Internet. Así mismo, cuando llega un paquete de Internet, sabe si es para *peque*, y se lo envía. Esto es lo que se conoce como IP Masquerading, o enmascaramiento de IPs y permite usar una sola dirección IP para conectar varias máquinas a la Red.

Todas estas tareas las lleva a cabo el kernel y es necesario que esté compilado con las opciones adecuadas. En concreto, deberá tener activadas las opciones del apartado "Networking Options":

- Network Firewalls
- TCP/IP Networking
- IP: forwarding/gatewaying
- IP: masquerading
- IP: ICMP masquerading
- IP: accounting
- IP: optimize as router not host

Una vez compilado e instalado el kernel y reinicializado el sistema, hay que indicar al kernel las reglas de reenvío. Esto se hace con la utilidad "*ipfwadm*" (IP firewall administration), añadiendo las dos líneas siguientes al script que inicializa la red:

```
ipfwadm -F -p deny
ipfwadm -F -a -m -S 10.10.10.0/24 -D 0.0.0.0/0
```

El IP masquerading funciona bien con las aplicaciones más usuales, como Web o correo, pero da problemas con el FTP, el IRC o el Quake. Para solucionar esto existen una serie de módulos como "*ip_masq_irc*" o "*ip_masq_ftp*".

5.4 Tarjetas de red (Ethernet)

Existen (al menos para el gran público) dos tipos de tarjetas Ethernet, según su velocidad. Las de 10 Mbps y las de 100 Mbps. La única diferencia está en la velocidad de trabajo. Muchas de las de 100 Mbps son compatibles con sus hermanas menores y se les suele llamar 10/100. Existen también estándares de Ethernet para comunicaciones a 1 Gb, pero ... \$\$\$

En cuanto al tipo de conexión física, pueden tener un conector tipo BNC para cable coaxial o un RJ-45 para un cable de par trenzado. También es posible que dispongan de los dos conectores. Es ese caso se las denomina tarjetas combo.

5.4.1 Instalar la tarjeta

Vale, hemos conseguido ahorrar 35 euros a base de pedir cubatas de Dyc en vez de Jack Daniels ¹ y podemos comprarnos una fastuosa tarjeta Ethernet. Solo queda instalarla en una ranura de expansión libre y configurar Linux para poder trabajar con ella.

Lo primero es identificar la tarjeta. Las más usuales suelen ser las compatibles con NE2000. Linux soporta gran número de tarjetas de red. Como en el caso de los protocolos PPP y PLIP, se puede recompilar el kernel para que la reconozca, o cargar un driver en forma de módulo.

Supongamos que nuestra tarjeta es una NE2000 o compatible. El comando para cargar el módulo sería:

```
insmod ne io=0x300 irq=7
```

- "ne" es el nombre del módulo correspondiente a la tarjeta
- "io=0x300" indica al kernel cuál es la dirección de entrada/salida que usa
- "irq=7" indica cuál es su número de IRQ. Con irq=0, se le pide que pruebe todas las IRQ hasta que encuentre la correcta. A veces funciona :)

¹Recomendamos otras formar de ahorro, naturalmente, pero éste fue un caso real. :-)

Lo activo como una conexión punto a punto entre jace y peque:

```
ifconfig plip0 10.10.10.11 pointopoint 10.10.10.12 arp up
```

Y añado una entrada a la tabla de enrutamiento, para que todos los paquetes dirigidos a cualquier dirección del tipo 10.10.10.xxx se envíen por el puerto paralelo:

```
route add -net 10.10.10.0 plip0
```

Los cambios hechos en la tabla de enrutamiento y en la configuración del interfaz de red desaparecen al reiniciar la máquina, por lo que es conveniente añadir las líneas anteriores a algún script que se ejecute cada vez que arranca el sistema (por ejemplo, `/etc/init.d/network` en la distribución Debian)

En jace ya está activo el enlace PLIP, pero ... ¿como se hace en MsDOS?. No hay problema, existe un "packet driver" para DOS que emula una tarjeta Ethernet. Este driver y gran cantidad de software de red para MsDOS lo podéis encontrar en cualquiera de las siguientes direcciones:

<http://www.agate.net/~tvdog>
<http://www.fdisk.com>

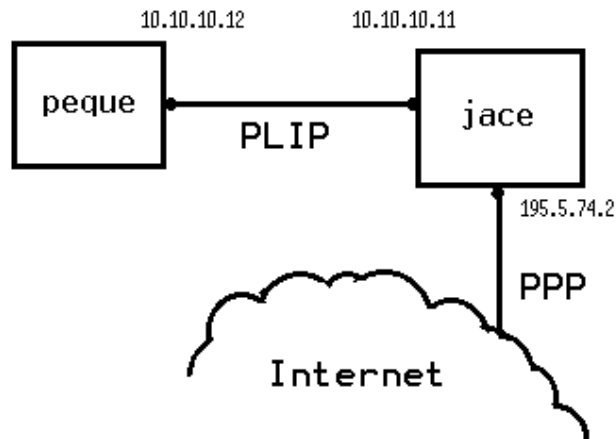
Para cargar el driver, solo hay que ejecutar `"PLIP 0x60"`

Al configurar los programas para DOS, usaremos de gateway y DNS a jace (10.10.10.11). Para que jace funcione de DNS es necesario que esté corriendo el demonio "named" y que esté convenientemente configurado.

IP Masquerading

Si lo que queremos es que cualquiera de los dos ordenadores pueda conectarse a Internet con el módem de *jace*, hay que hacer unos ajustes.

Cuando se establece la conexión, el proveedor asigna a jace una dirección IP, por ejemplo 195.5.74.2. Esta es la dirección del interfaz ppp0 de *jace*. *Peque* no puede acceder a Internet, porque no tiene una dirección IP de Internet, 10.10.10.12 es una dirección de nuestra red interna y no puede usarse para salir fuera.



- Y el último "truco" consiste en hacer que tu sistema arranque directamente en modo gráfico... esto es algo que algunos "winusers" echan en falta cuando se les habla de LINUX... que "es en modo texto"!!! Nada mas lejos de la realidad! Antes de nada, porque esto tiene "cierto riesgo" haz "man init" y lee la documentación ¿ok?

Ahora ya estás listo para el "truco", pero ahora sabiendo como funciona. :-). Estando en una sesión de administrador del sistema (root) teclea "init 3". Esto hará que a partir de ese momento, la máquina entre en modo gráfico. Para devolver la máquina al original, como administrador pulsa "init 2" y volverás a modo texto. Mucho cuidado con los "init's" porque son diferentes estados de la máquina y si los usas sin saber que hacen puedes tener algún problema ¿ok? :-)

3.7 ¿Necesitas algo más.? Prográmalo tu mismo!!! :-P

Hay aun MUCHAS cosas por hacer dentro del mundo LINUX... Posiblemente tengas alguna idea en mente, eches de menos alguna aplicación... Pues si sabes programar y te animas... prográmala! O Únete a algún equipo de desarrolladores de software o... ayuda a traducir el software ya existente al castellano, o la documentación (HOWTOS ==¿ COMOS), las páginas del manual "man". Si te animas a alguna cosa de estas... pásate por algunas de las URL's que vienen en el apéndice de direcciones de interés.

5.3.1 Conseguir el cable

Se puede buscar en tiendas de informática o electrónica con el nombre de cable Laplink paralelo, Turbo Laplink o null-paralelo. También se puede hacer en casa, con las herramientas adecuadas y buena mano. Las conexiones entre los dos conectores DB-25 son las siguientes:

```
1 - 1
2 - 15
3 - 13
4 - 12
5 - 10
6 - 11
10 - 5
11 - 6
12 - 4
13 - 3
14 - 14
15 - 2
16 - 16
17 - 17
25 - 25
(tierra)
```

El resto de los pines no se conectan. El cable puede alcanzar una longitud de 15 metros como máximo.

5.3.2 Instalación de PLIP

El protocolo PLIP, al igual que el PPP, se puede incorporar al compilar el kernel o bien cargarse como módulo. Para cargar PLIP como módulo hay que ejecutar `insmod plip`.

La instalación del módulo puede fallar por dos razones:

1. Esta cargado el módulo "lp" que controla el puerto paralelo. Hay que desinstalar el módulo o recompilar el kernel para que no cargue el soporte para el puerto de impresora.
2. La dirección e/s o la IRQ del puerto no son estándar. Las direcciones de entrada/salida e IRQs que toma Linux por defecto son:

Interfaz	I/O	IRQ
plip0	0x3bc	7
plip1	0x378	7
plip2	0x278	5

Para instalar plip con otros parámetros hay que añadir las opciones correspondientes, por ejemplo: `insmod plip io=0x3fe irq=3`. Si todo va bien, ya hay un interfaz nuevo al que dirigir los paquetes IP. Con PLIP se puede establecer una conexión con otro Linux o con una máquina en MsDOS. No es posible conectar con una máquina en Windows 95/98, ya que nadie se ha molestado en programar un controlador PLIP para la pila de protocolos de este sistema operativo.

5.3.3 Un ejemplo real

En casa tengo dos ordenadores, un Pentium 100 con Linux, al que llamaré "jace" y un 386 con MsDOS, "peque". Quiero conectar los dos por el puerto paralelo para usar a "peque" como terminal de "jace" y poder acceder a su disco duro. Doy a cada máquina una dirección IP y las incluyo en el `/etc/hosts` de jace, que queda como sigue:

```
10.10.10.11 jace
10.10.10.12 peque
```

Ahora configuro el interfaz de red plip0 con la nueva dirección IP:

```
ifconfig plip0 10.10.10.11
```



```
ABORT
"NO DIALTONE"
```

```
ABORT
"NO ANSWER"
```

```
" "
ATZ
```

```
OK
ATDT901505055
```

```
CONNECT
\d\c
```

Si la versión del "pppd" es anterior a la 2.3.3, es necesario cambiar la línea "user 'linuxero@arrakis'" por "+ua /etc/ppp/linuxero" y crear el archivo "/etc/ppp/linuxero" con el nombre de usuario en la primera línea y la contraseña en la siguiente. También es necesario indicar a Linux la dirección del servidor DNS, para que pueda convertir nombres de hosts como "www.yahoo.com" en direcciones IP. Todo lo relativo a la resolución de nombres se encuentra el fichero "/etc/resolv.conf"

```
nameserver 195.5.64.2
nameserver 62.81.31.250
domain arrakis.es
```

La tercera línea es opcional. Sirve para añadir por defecto el dominio "arrakis.es" a los nombres. Por ejemplo, si tecleamos "ping chusuck", Linux lo interpretará como "ping chusuk.arrakis.es".

Una vez creados los ficheros anteriores, ya nos podemos conectar ejecutando, como root:

```
pppd call arrakis
```

Si todo va bien el módem comenzará a marcar, se conectará con el servidor y se negociará la conexión. Se puede monitorizar todo este proceso examinando el log de mensajes con:

```
tail -f /var/log/messages
```

Una forma de verificar el correcto funcionamiento de la red es hacer un "ping" a una máquina remota. Si ejecutamos "ping www.arrakis.es", la respuesta debería ser algo como:

```
PING www.arrakis.es (195.5.83.2): 56 data bytes
64 bytes from 195.5.83.2: icmp_seq=0 ttl=64 time=123.2 ms
64 bytes from 195.5.83.2: icmp_seq=1 ttl=64 time=108.9 ms
64 bytes from 195.5.83.2: icmp_seq=3 ttl=64 time=210.1 ms
(...)
```

Si la conexión a la Red no se ha realizado con éxito, aparecerá algún mensaje como "No route to host", "host not responding" o simplemente se quedará mudo después de la primera línea.

Para desconectarse de La Red se puede matar el proceso del demonio PPP con "killall pppd" o, aún peor, desenchufar el módem si es externo.

5.3 PLIP, una red para pobres

PLIP son las siglas de Parallel Line IP. Es un protocolo que emula una conexión ethernet por el puerto paralelo.

Como dije al principio, la mejor forma de conectar dos ordenadores en red es usando un par de tarjetas ethernet y un pedazo de cable. Una tarjeta ethernet viene a costar unas 2.500 pts., así que por mil duros (poco más de 30 euros) se pueden compartir los recursos de las dos máquinas que tenéis en casa.

Otra solución es comprar un cable paralelo tipo "LapLink" y usar el puerto paralelo para conectarlas. La ventaja es el precio (1.100 pts. en la tienda más cara de Madrid), el inconveniente es la velocidad, que oscila entre los 10 y los 30 Kbytes/s frente a los 1.25 Mbytes/seg de las ethernet.

Capítulo 4

Introducción a las redes TCP/IP

Hoy en día las redes de ordenadores, tanto locales como mundiales, han cambiado el panorama de la informática. Ya están prácticamente olvidados los días donde se contemplaba el ordenador como una máquina aislada, siendo el presente un mundo donde la información fluye a través de los cables telefónicos, ópticos, satélites y demás medios. Actualmente se hace realidad ese lema de Sun Microsystems: "El ordenador es la red". Y es en este mundo, el "cyberespacio" como algunos lo denominan, donde el sistema operativo Linux muestra una clara superioridad sobre otros sistemas (tales como Windows). Y es que Linux ofrece multitud de soluciones y posibilidades en el contexto de las redes informáticas, siguiendo la tradición de los clásicos sistemas Unix.

4.1 Enredando

Desde un punto de vista práctico, los usuarios de ordenadores normalmente dividen el mundo de las redes informáticas en dos mitades

Redes de Área Local (LANs) son redes de alta velocidad usadas para conectar ordenadores en una misma localización. Los típicos ejemplos de redes LAN están basados en Ethernet, Token Ring o 10Base-T (también conocido como par trenzado). Las LAN alcanzan actualmente velocidades entre 10 y 100 megabits/seg, aunque se están dando avances muy grandes para conseguir mayores velocidades.

Redes de Área Extensa (WANs) son redes por lo general más lentas¹ que las LAN, y son utilizadas por las organizaciones para conectar entre sí a varias LAN situadas en diferentes puntos geográficos. Este tipo de conexiones puede hacerse a través de líneas telefónicas, líneas digitales, etc.

4.1.1 Internet

Una de las primeras redes de ordenadores fue la ARPANET, desarrollada a mediados de los años 70 por universidades y empresas que trabajaban bajo el contrato de la sección ARPA² del departamento de defensa americano. La ARPANET enlazaba ordenadores a través de todo el mundo y servía como parte central de otras redes de empresas y universidades. Más tarde el departamento de defensa dejó el proyecto y todo derivó a lo que hoy es Internet.

Internet es una red basada en el protocolo IP (Internet Protocol), y enlaza varios millones de ordenadores y a cientos de millones de usuarios de todo el mundo. Todos estos usuarios de la red pueden utilizar la infraestructura de ésta para compartir información y comunicarse.

4.2 Unix y las redes de ordenadores

Los sistemas operativos Unix se han beneficiado y a la vez han contribuido mucho a la popularidad de las redes. Cuando en 1.983 se lanzó el sistema Berkeley 4.2, éste trajo consigo una implementación bastante buena y muy fiable del protocolo

¹Esto se cumple para la gran mayoría de casos, pero de hecho algunas WAN alcanzan velocidades de hasta 45 Megabytes/segundo (T3).

²Advanced Research Projects Agency (Agencia para Proyectos de Investigación Avanzados).

IP. Desde entonces, el código de comunicaciones de Berkeley ha sido adoptado por la mayoría de fabricantes de sistemas Unix, así como por otro tipo de sistemas no basados en Unix.

Después de más de una década de desarrollo, los sistemas operativos Unix han evolucionado a un punto tal que prácticamente todas las cosas que puedes hacer con un gran ordenador se pueden realizar igual o mejor con una red de estaciones Unix. Ejemplos:

- Terminales remotas virtuales (telnet y rlogin): te permiten conectarte dentro de una máquina de la red.
- Servicio de archivos remoto: puedes acceder a los archivos de otro ordenador mientras te hayas en otro.
- Correo electrónico: con el que puedes mandar mensajes a usuarios de otros ordenadores.
- Servicios de información (finger, whois, etc): te permite acceder a información sobre un usuario de un ordenador (tales como su teléfono, dirección, ...).
- Fecha y hora: permite a tu ordenador que se sincronice automáticamente con el resto de ordenadores de la red.
- Llamadas de proceso remoto (RPC): con las que puedes ejecutar programas y subrutinas que sistemas remotos como si estuvieran en el propio.

4.2.1 Linux en red

Linux, como sistema operativo Unix derivado del BSD, se beneficia de todas las ventajas de los sistemas Unix, y es más, en combinación con herramientas de otro tipo, se alcanzan rendimientos iguales o superiores a los llamados "Sistemas Unix comerciales".

Además, encontramos que Linux posee soporte para una gran cantidad de hardware de comunicaciones, que va desde lo mas normal (módems, tarjetas ethernet ...), hasta elementos más extraños (Soporte AX25 para radio, ...). Tampoco hay que olvidar que si no existiera el driver necesario para un determinado hardware, la comunidad Linux podría encargarse de escribir uno o incluso el mismo usuario, si está capacitado para ello, hacerse el suyo propio.

Es esta extraordinaria capacidad para trabajar en red la que está causando que Linux aumente su presencia como servidor en Internet, donde su uso, junto con programas como Apache³ está desbancando a otro tipo de sistemas tales como Windows NT.

4.2.2 Interfaces de red

Cuando nos conectamos a una red, de cualquier tipo, lo hacemos mediante un hardware especial para ello. Así, podemos utilizar módems, tarjetas ethernet, cables de puerto paralelo, etc. A cada uno de estos elementos que utilizamos les llamamos "interfaz de red", y puede haber varios en una misma máquina.

Linux, para tratar con cada dispositivo, asigna un nombre a cada interfaz de red. Este nombre dependerá del tipo de hardware con el que tratemos. Básicamente cada interfaz tiene un nombre y un número. Esto es, imaginemos un ordenador que tiene 3 módems y dos tarjetas ethernet:

1. ppp0: primer módem
2. ppp1: segundo módem
3. ppp2: tercer módem
4. eth0: primera tarjeta ethernet
5. eth1: segunda tarjeta ethernet

Como se puede ver, el número es cero para el primero, uno para el segundo y así. El nombre es eth para tarjetas ethernet y ppp para módem. Para ethernet el nombre es evidente, el nombre de "ppp" es porque se refiere a una conexión punto a punto (ppp = "Point to Point Protocol").

Hay más tipos de interfaces, pero estos son los más usuales.

Adicionalmente hay una interfaz adicional llamada *loopback*, que se usa para referirse a la misma máquina en la que nos encontramos aparece en las tablas como *lo*.

³El programa Apache es un servidor Web gratuito de extraordinario rendimiento. Para más información <http://www.apache.org>

```
/sbin/isapnp /etc/isapnp.conf (si el módem es PnP)
/bin/setserial <opciones>
```

Llegados a este punto hay que hablar de los llamados "Winmodem", "Voicemodem" o "Módems HSP". Se trata de modems internos de bajo coste que no funcionan en Linux, solo son capaces de trabajar en Windows con un controlador específico. La razón es que necesitan tiempo de CPU para MODular y DEModular, no son más que simples conversores A/D - D/A, el resto del trabajo se hace por software. Se puede empezar a sospechar si en la caja pone como requisito que necesitamos mínimo un Pentium, alguna cantidad de RAM o que el sistema operativo sea Windows.

5.2.2 Configuración del demonio pppd

Las conexiones por módem a una red TCP/IP se pueden establecer con dos protocolos: PPP y SLIP. Aquí se va a describir el método de conexión mediante PPP ya que SLIP ha quedado obsoleto. PPP son las siglas de Point to Point Protocol, o Protocolo Punto a Punto. Se utiliza para realizar comunicaciones TCP/IP a través de una línea serie, como es la telefónica.

Para que Linux pueda usar PPP, el protocolo debe estar incorporado en el kernel o cargarse en forma de módulo. Para saber si PPP esta cargado sólo hay que revisar los mensajes que da el kernel al arrancar y buscar algo del tipo:

- PPP: version 2.2.0 (dynamic channel allocation)
- PPP: Dynamic chanel allocation code copyright 1995 Caldera, Inc.
- PPP: line discipline registered.

Para cargar ppp como módulo sólo hay que teclear "*insmod ppp*". Si se va a usar ppp de manera muy a menudo es más eficiente compilarlo en el kernel.

Con el módem configurado y el protocolo PPP en funcionamiento, solo resta conectarse a la Red. Lo primero es decirle a "pppd" cual es la contraseña que debe usar. Hay que editar los ficheros "*/etc/ppp/chap-secrets*" y "*/etc/ppp/pap-secrets*" y añadir la línea:

```
'linuxero@arrakis' * linux123
```

Teniendo en cuenta que las separaciones no son espacios, sino tabuladores. En realidad no es necesario editar los dos ficheros, solo el que corresponda al tipo de negociación que use nuestro servidor (PAP o CHAP). Infovía Plus y Retenet usan el protocolo de autenticación CHAP. Teniendo en cuenta que esos dos ficheros contienen nuestra clave descriptada, es altamente recomendable que solo puedan ser leídos por root. Eso se hace con "*chmod 600 ???-secrets*".

Ahora se crean los fichero de opciones "*/etc/ppp/peers/arrakis*" y "*/etc/ppp/arrakischat*":

```
noauth
connect "/usr/sbin/chat -v -f /etc/ppp/arrakischat\char"
```

```
debug
/dev/ttyS1
115200
defaulroute
noipdefault
user 'linuxero@arrakis'
kdebug
2
```

```
ABORT
EUSY
```

```
ABORT
"NO CARRIER"
```

```
ABORT
VOICE
```

- Usuario: "linuxero@arrakis"
- Password: "linux123"
- N° de teléfono: 901505055
- Un módem conectado en el puerto COM2

Los datos anteriores corresponden a una conexión a Arrakis, por Infovía Plus (un ejemplo bastante típico de conexión), habrá que modificarlos para que concuerden con la configuración específica de cada uno. Por supuesto, el usuario y la contraseña son falsos. El DNS secundario es el de Iddeo-Retenet.

5.2.1 Configuración del puerto serie

Antes de nada hay que identificar y configurar adecuadamente el puerto serie a través del cual se va a realizar la conexión. Si el módem es externo no hay mayor problema, ya que se conecta al puerto del ordenador y es fácilmente identificable. Bajo Linux el llamado com1 de Windows/DOS corresponde al dispositivo /dev/ttyS0; com2 al ttyS1, y así sucesivamente.

En el caso de que el módem sea interno, el puerto serie viene incorporado en el aparato, por lo que es necesario saber cual es, que IRQ y que dirección de entrada/salida usa. Para ello resulta de gran ayuda examinar "/proc/interrupts" y "/proc/ioports" o echar un vistazo a la configuración de Windows, si está a mano (no solo de Linux vive el hombre). Los parámetros estándar que toma Linux por defecto son:

DOS	Linux	IO/PORT	IRQ
com1	/dev/ttyS0	0x03f8	4
com2	/dev/ttyS1	0x02f8	3
com3	/dev/ttyS2	0x03e8	4
com4	/dev/ttyS3	0x02e8	3

Si el puerto del módem no usa una configuración estándar, habrá que especificárselo al kernel con el comando "setserial". Por ejemplo, si en puerto serie "com4" tiene la dirección de entrada salida en "0x2f8" y usa la petición de interrupción número 5, habrá que teclear (como root):

```
setserial /dev/ttyS3 irq 5 port 0x2f8 skip_test spd_vhi autoconfig
```

En el caso en el que el módem sea PnP (Plug and Pray ... digo ... Play) tenemos dos opciones:

- Si la BIOS es PnP, se puede especificar allí la IRQ y la dirección que luego se pasarán al setserial
- Si no es así, se deberán usar las "ISAPnP Tools", que suelen venir con la mayoría de las distribuciones de Linux

En este último caso los pasos a seguir serían:

- Ejecutar el comando "isapnpdump ¿ /etc/isapnp.conf"
- Editar el fichero "isapnp.conf" descomentando aquello que interese de la IRQ y el puerto, para que quede de la forma:

```
# Card 1: (serial identifier 8a 00 09 7c a6 81 13 b0 4e)
#
ANSI string --> SupraExpress 336i PnP Voice Módem <--
(CONFIGURE SUP1381/621734 (LD o (IO O (BASE 0x02e8))
(INT O (IRQ 3 (MODE +E)))) (ACT Y)))
```

- Ejecutar el comando "isapnp /etc/isapnp.conf" para que el dispositivo sea reconocido donde se le ha dicho.
- Ejecutar el "setserial" como se mencionó arriba.

Una vez configurado el puerto, se puede comprobar su correcto funcionamiento con el programa "minicom" o con cualquier otro terminal. Basta con lanzarlo como root y teclear "AT". Si se recibe un "OK", es que está funcionando adecuadamente.

La configuración del puerto se perderá cada vez que se reinicie el sistema, así que lo más recomendable es añadir a los scripts de inicio (por ejemplo "etc/rc.d/rc.local" en RedHat o "etc/rc.boot/0setserial" en Debian) las líneas:

4.2.3 Modelo cliente-servidor

En el mundo de TCP/IP las comunicaciones entre ordenadores se rigen básicamente por lo que se llama *modelo cliente-servidor*, este es un modelo que funciona extraordinariamente bien y es responsable de la flexibilidad en las comunicaciones.

Su funcionamiento es sencillo: nosotros tenemos una máquina *cliente*, que requiere un servicio de una máquina *servidor*, y éste realiza la función para la que está programado (nótese que no tienen que tratarse de máquinas diferentes).

Entrando más en detalle en lo que a Linux/Unix se refiere, los programas servidores⁴ se ponen en lo que se denominan *puertos*, a los que nos referiremos con un número. Podemos ver al puerto como una especie de amarre donde los programas residen esperando a ser utilizados.

Cuando el cliente realiza una petición al servidor, el origen de la conexión tiene un *puerto origen*, al que también nos referimos con un número. El servidor recibe la petición y realiza una copia de sí mismo que atenderá al cliente, quedando otra copia en el puerto original para atender otras peticiones⁵. Veamos como funciona en un ejemplo:

Un usuario desea recoger su correo de la máquina de su proveedor de internet. Cuando el usuario da la orden a su programa de correo, éste se conecta desde un puerto arbitrario suyo (por ejemplo el 12023) al puerto 110 de la máquina que se encarga del correo (que suele ser lo usual). A partir de ahora toda la comunicación entre el programa de correo (cliente) y el servidor de correo, fluirá por una tubería imaginaria entre ambas máquinas, siendo los extremos de dicha tubería los puertos de origen y destino.

4.2.4 Direcciones de Internet

A cada interfaz que un ordenador tenga en una red IP (como Internet) se le asigna una única dirección de 32 bits. Estas direcciones se suelen expresar como cuatro números de 8 bits separados por puntos (ej: 193.146.25.2). Pensemos acerca de una dirección IP como si fuera un número de teléfono; si conocemos la dirección IP de un ordenador, podemos conectarnos a él para intercambiar información. Además, en una subred nos referiremos a los diferentes ordenadores por direcciones IP que previamente habremos asignado.

4.2.5 Protocolos

Para comunicar diferentes máquinas tenemos que acordar la forma en que lo vamos a hacer. Al conjunto de normas para la transmisión de mensajes en una red se le llama *protocolo de red*, en el caso que nos ocupa que es Internet hay varios, entre los más importantes:

IP Es el responsable de que los paquetes que mandamos desde un sitio lleguen a su destino. Se encarga del enrutamiento de los paquetes de información.

TCP y UDP Son dos protocolos que necesitan de IP para transmitirse (se dice que están en otra capa). TCP se encarga de transmitir información de forma segura, asegurándose de que el mensaje llega y se encarga de retransmitirlo si no va así (*stream*). UDP al contrario se limita a enviar el paquete (aquí llamado *datagrama*), sin preocuparse si llega a su destino. Aunque parezca ilógico, UDP es necesario para cierto tipo de tareas.

ICMP Son mensajes de control de la red. Este protocolo se utiliza para mandar mensajes especiales, tales como para ver si una máquina se encuentra conectada, si su reloj está sincronizado, y para mandar algunos mensajes de error.

4.2.6 DNS

Cuando hemos visto las direcciones IP, hemos observado que no son más que números. Cuando intentamos conectar con un ordenador, suele ser más fácil referirnos a él por un nombre conocido. Seguramente al lector le sea familiar ver nombres como ulises.uax.es, www.yahoo.com, etc. Estos nombres son direcciones de red y como tal tienen una dirección IP, pero nos podemos referir a esos ordenadores por este nombre. En algún momento se tiene que hacer la conversión de nombre a número, es de éste asunto de lo que se encargan los servidores de DNS.

Los servidores de DNS están distribuidos piramidalmente: hay unos servidores primarios (importantísimos para el funcionamiento de Internet), y colgando de ellos una sucesión de servidores secundarios, terciarios, ...

Cuando nosotros escribimos por ejemplo *www.uax.es* en un navegador, éste consulta en la configuración de red cual es el servidor de DNS que debe de utilizar, y le preguntará a ese que número le corresponde a ese nombre. Si no lo

⁴A veces a estos programas se les llama *demonios* o *daemons*.

⁵Quien lo desee puede consultar la bibliografía para hallar una explicación más rigurosa.

sabe, ese servidor preguntará a uno de los servidores superiores (con una mayor base de datos), hasta que lo encuentre y entonces devolverá el número. Si no lo encontrara, se le presentará al usuario un mensaje de que la dirección introducida es inexistente.

Capítulo 5

Configurando el interfaz de red

Para empezar a trabajar directamente con redes bajo Linux, tendremos que configurar nuestro interfaz de red, ya que si no, no podremos acceder al medio físico por el que se moverá la información. Aunque los tipos de conexión son muchos, en este capítulo cubriremos los más habituales: configuración de un módem para conectar con Infovia Plus (la mayor puerta de acceso a Internet en España), y configuración de una tarjeta Ethernet. Además veremos como realizar una conexión a través del puerto paralelo entre dos ordenadores para, de este modo, poder realizar una pequeña intranet en nuestra casa, o aprovechar un equipo poco potente como terminal Unix. Además haremos una introducción a la conexión de un sistema de radio por el protocolo AX25.

5.1 Introducción

Linux es un sistema operativo diseñado para trabajar en red. Dispone de todas las aplicaciones imaginables, tanto clientes como servidores, para sacar el máximo partido a la máquina cuando se encuentra bien en una red interna, o bien en la Red de redes.

Pero no todos tenemos la suerte de contar con una red en nuestra casa. El caso más típico es un ordenador windows-multimedia solitario, condenado a servir de máquina recreativa o de escribir. En este capítulo, se describirá como conectar el ordenador a una red, de tres formas distintas:

- Con un módem, con el protocolo **PPP**
- Con un cable serie, con el protocolo **PLIP**
- Con una tarjeta de red tipo **Ethernet**
- A través del protocolo **AX25**

5.2 PPP. Una ventana abierta a la Red

En Windows 95/98 conectarse a Internet con un módem no supone mayor problema. Sólo hay que introducir el CD-ROM que te da el proveedor de Internet, esperar a que actúe la "reproducción automática" y pinchar en el botoncito "Instalar". Y si no tenemos el CD, se puede llamar al número 900 de la compañía, o al 900505055 de Infovia y un lorito repetirá de carrerilla donde hay que pinchar con el ratón para configurar el "Acceso telefónico a redes".

Pero esto es Linux. Lo más probable si se llama al número de asistencia técnica del proveedor preguntado cómo conectarse con Linux, es que la respuesta sea "¿eing? ¿linsus, que es linsus?". En fin ...

De ahora en adelante, se supondrán unos conocimientos mínimos del sistema operativo. Nada del otro mundo, cómo editar un fichero, moverse por los directorios y cosas así. También se supondrá que tenemos los los siguientes datos, que los debería proporcionar el nuestro servidor.

- DNS Primario: 195.5.64.2
- DNS Secundario: 62.81.31.250