MANUALES

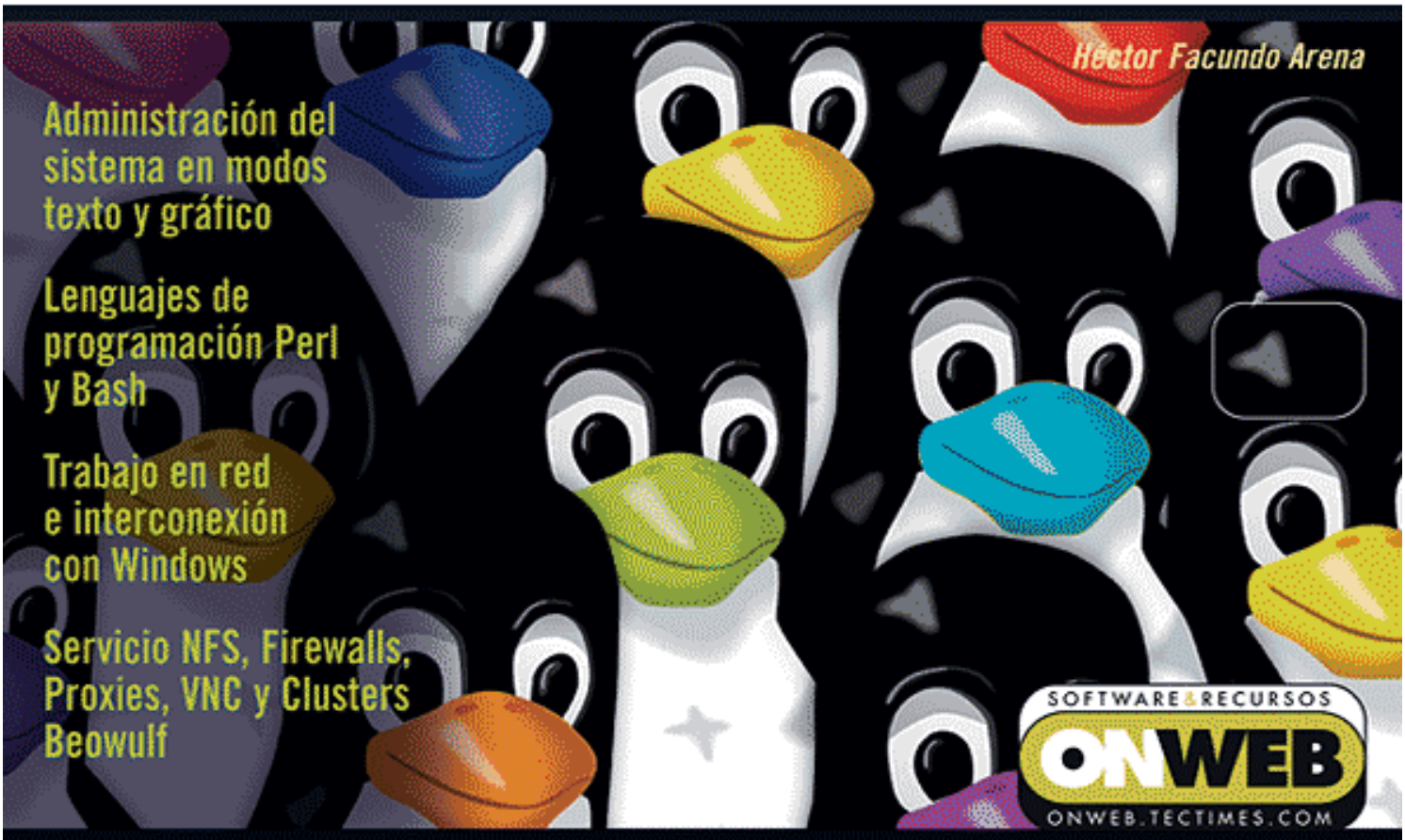
USERS

TU PUERTA DE ACCESO AL MUNDO DIGITAL

LA BIBLIA DE Linux

Manual de uso, instalación y configuración

Héctor Facundo Arena



Administración del sistema en modos texto y gráfico

Lenguajes de programación Perl y Bash

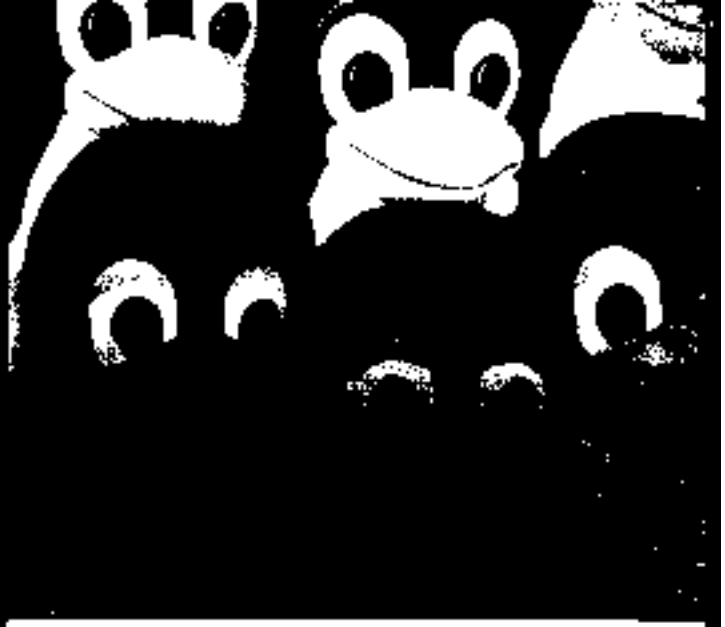
Trabajo en red e interconexión con Windows

Servicio NFS, Firewalls, Proxies, VNC y Clusters Beowulf

SOFTWARE & RECURSOS
ONWEB
ONWEB.TECTIMES.COM

Aprenda a montar y a configurar servidores web

MP



TÍTULO: LA BIBLIA DE LINUX

AUTOR: Hector Facundo Arena

COLECCIÓN: Manuales USERS

EDITORIAL: MP Ediciones

FORMATO: 17 x 24 cm

CANTIDAD DE PÁGINAS 264

Editada por MP Ediciones S.A., Moreno 2062 (C1094ABT),
Ciudad de Buenos Aires, Argentina
Tel: (54 11) 4959 5003 Fax: (54 11) 4954-1791

ISBN 987-526-020-3

Copyright © MP Ediciones S.A. 2002
Hecho el depósito que marca la ley
Reservados todos los derechos de autor
Prohibida la reproducción total o parcial de esta publicación por cualquier medio o procedimiento y con cualquier destino

Primera impresión realizada en enero de 2003
New Press Grupo Impresor S.A.,
Paraguay 264, Avellaneda,
Provincia de Buenos Aires, Argentina

Todas las marcas mencionadas en este libro son propiedad exclusiva de sus respectivos dueños

LA BIBLIA DE LINUX

Manual de uso, instalación y configuración

Héctor Facundo Arena

Sobre el autor

Héctor Facundo Arena es consultor de temas relativos a GNU/Linux y otras aplicaciones de Software Libre desde 1998. A lo largo de su carrera, editó cinco libros, lideró varios proyectos de desarrollo relacionados con el sistema y sirvió de consultor a usuarios, empresas y organismos gubernamentales. Fue el editor principal de la primera revista de GNU/Linux de la Argentina (Linux USERS), y colabora en la sección Linux de la revista USERS. Además, es director de su propio centro de capacitación en tecnologías de Software Libre (Tuxsys), y ha dictado decenas de cursos y seminarios en diferentes puntos del país. Héctor es miembro oficial del proyecto GNU, y puede ser localizado en su dirección de correo electrónico, **hf@gnu.org**.

Dedicatoria

A mi padrino, César Ioppolo.

Agradecimientos

A mis padres, por apoyarme en todos mis proyectos; a Jorge Fajardo, por haberme propuesto la idea de este libro; a Miguel Lederkremer, por haber confiado en mi desde mis comienzos; y a mis lectores, ya que, sin ustedes, nada de esto tendría sentido.

Sobre la editorial

MP Ediciones S.A. es una editorial especializada en temas de tecnología (computación, IT, telecomunicaciones).
Entre nuestros productos encontrará: revistas, libros, fascículos, sitios en Internet y eventos.
Nuestras principales marcas son: *USERS, Aprendiendo PC* y *TecTimes*.
Si desea obtener más información, puede contactarnos de las siguientes maneras:
Sitio web: www.tectimes.com
E-mail: libros@tectimes.com
Correo: Moreno 2062 (C1094ABF), Ciudad de Buenos Aires, Argentina.
Tel.: 54-11-4959-5000 / **Fax:** 54-11-4954-1791

Prólogo

Desde que escribí mi primer libro, a fines del año 1999, hasta la actualidad, la situación general del mercado tecnológico ha cambiado notablemente. Dentro de ese contexto de permanentes cambios, la aceptación que ha tenido el sistema operativo GNU/Linux en Latinoamérica es realmente asombrosa, ya que muchas empresas, escuelas y organismos gubernamentales han comenzado a implementarlo de forma exitosa.

En estos años, mi vida también ha cambiado notoriamente. He tenido la oportunidad de enseñar el uso del sistema operativo a centenares de personas, de viajar por diferentes regiones dando charlas y conferencias sobre los más diversos temas, de participar en reuniones junto a altos funcionarios de grandes empresas interesadas en implementarlo, y he tenido la oportunidad de dirigir la primera revista de GNU/Linux en la Argentina: Linux USERS.

Todas esas experiencias dejaron algo en mí, y puedo asegurarles que he vivido en piel propia el crecimiento que ha desarrollado el sistema operativo por estos rincones. Crecimiento que parece seguir evolucionando día a día.

Realmente no sé si GNU/Linux es el sistema operativo del futuro. Creo que nunca lo sabré, y creo también que nadie tiene las razones necesarias y suficientes para saberlo. De todas formas, hoy por hoy, GNU/Linux es el mejor sistema operativo para la plataforma PC.

Este libro significó un gran proyecto para mí. El nombre ya me obligó, desde un comienzo, a cubrir detalladamente los temas más importantes relacionados con el sistema operativo. Aquí encontrarán una recopilación de los temas tratados en tres de mis cuatro libros anteriores (*Linux Avanzado*, *Linux Fácil* y *Sitios web bajo Linux*), así como una selección de mis mejores artículos para las revistas y los diferentes sitios de Internet en los que he participado.

Creo que el nombre del libro hace honor a su contenido, y viceversa.
He tratado de abarcar el más amplio espectro de temas, desde descripciones de los diferentes directorios en el sistema de archivos, hasta la instalación de un firewall, pasando por clusters, servidores web, programación e interconexión con otros sistemas operativos.

Para la selección de temas, me basé en mis años de experiencia al frente de cursos de capacitación y de las centenas de e-mails que recibo mensualmente con consultas y sugerencias. Siempre he intentado estar cerca del usuario y enterarme de cuáles son sus necesidades. Por eso, la idea del libro es tratar de servir como referencia básica de

todo usuario de GNU/Linux, desde el más novicio, que desea dar sus primeros pasos en la materia, hasta el más experimentado.

Algunos quizá se pregunten por qué hay tan poco espacio dedicado a las aplicaciones gráficas, multimediales y al entorno gráfico en general. Los que me conocen saben que tengo una filosofía muy particular para impartir conceptos: "Si se comprende el funcionamiento de un procedimiento en el modo texto, entonces el modo gráfico será mucho más fácil de aprender".

Esta filosofía la vengo aplicando desde mi primer libro, y ha pasado por todos mis cursos y seminarios, y mis lectores y alumnos (a la larga) terminan siempre agradecidos.

Además, recordemos el nombre del libro. En La Biblia original no se habla de autos ni motores, por ejemplo. En cambio, se hace mención del concepto de transporte y la necesidad de herramientas que ayuden al hombre. En este libro, la idea es exactamente la misma: aquí encontrarán los temas más profundos del sistema operativo, y generalmente estos temas son gestionados desde el entorno textual, intercalados con conceptos filosóficos acerca del desarrollo y la distribución de éste.

Sólo me queda agradecerles la atención que me han prestado durante todos estos años y desearles el mayor de los éxitos en su camino junto al sistema operativo GNU/Linux. Disfruten de esta Biblia. La Revolución ya está aquí.

Héctor Facundo Arena

El libro de un vistazo

A continuación presentamos un resumen orientador del contenido del libro, que les permitirá conocer rápidamente los temas tratados en cada uno de los capítulos.
Como se puede apreciar, la obra contempla los más diversos temas, desde los conceptos básicos, para quienes no tienen experiencia y desean dar sus primeros pasos con el sistema operativo, hasta los temas más avanzados, para quienes sí tienen experiencia en la materia pero quieren encontrar conceptos que les sean de utilidad, y técnicas para solucionar diferentes tipos de problemas.

CAPÍTULO 1

Génesis

Este capítulo, básicamente teórico, tiene como objetivo principal aclarar todos los conceptos introductorios y hasta filosóficos que giran alrededor de la expresión GNU/Linux.

CAPÍTULO 2

Primeros pasos

Con la ayuda de este capítulo, daremos nuestros primeros pasos en el uso de GNU/Linux; por ejemplo, analizaremos los métodos utilizados para registrarse, trabajar y apagar el sistema.

CAPÍTULO 3

Trabajando en el modo texto

En este capítulo aprenderán los conceptos básicos para manejar un sistema GNU/Linux desde el entorno textual. Olvídense del mouse por un rato y ¡a teclear!

CAPÍTULO 4

La programación en el lenguaje Bash

El intérprete de comandos Bash incluye un lenguaje de programación de scripts muy poderoso. Analizaremos su uso en este capítulo.

CAPÍTULO 5

La programación en lenguaje Perl

Perl es uno de los lenguajes más famosos de estos tiempos. Su facilidad de uso y su velocidad de procesamiento lo convierten en el más apto para el desarrollo de aplicaciones CGI para la Web.

CAPÍTULO 6

El modo gráfico

En este capítulo vamos a dejar de teclear un poco para internarnos en el mundo gráfico de GNU/Linux, y trabajar fundamentalmente con la configuración de entornos de trabajo.

CAPÍTULO 7

GNU/Linux en red

Antes de comenzar a ver los diferentes servicios de red, vamos a configurar nuestros sistemas. Luego haremos uso de algunos servicios del protocolo TCP/IP.

CAPÍTULO 8

Login remoto

Comenzaremos esta serie de capítulos dedicados a los entornos de redes hablando de un servicio sumamente útil: el login remoto.

CAPÍTULO 9

Instalar un servidor web

Apache es uno de los servidores web más populares para Internet. En este capitulo, aprenderemos a instalarlo y a configurarlo.

CAPÍTULO 10

El servicio NFS

Mediante el servicio de compartimiento de archivos NFS (Network File System) podremos transferir archivos a través de una red de manera rápida y transparente.

CAPÍTULO 11

Interconexión en red con Windows™

En este capitulo analizaremos el sistema Samba, el cual nos permite compartir archivos entre sistemas Windows y GNU/Linux de manera totalmente transparente para el usuario.

CAPÍTULO 12

Firewalls y proxies

El sistema IPTables incluido en la serie 2.4 del kernel Linux nos permite definir firewalls y proxies de forma sumamente práctica.

CAPÍTULO 13

El sistema VNC

Anteriormente hablamos de acceso remoto a través de terminales textuales. En este capitulo nos referiremos al control remoto de interfaces gráficas.

CAPÍTULO 14

Clusters Beowulf

Aquí analizaremos el método que permite que muchas computadoras de poca potencia trabajen en conjunto para formar entre ellas un supersistema de cómputo.

APÉNDICES

Documentación oficial

En el apéndice A se incluyela conocida GNU Public License (GPL), versión 2 (de junio de 1991), que cubre la mayor parte del software de la Free Software Foundation, y muchos más programas.

Además, el apéndice B presenta otro documento oficial de la comunidad, denominado "Abogacía por Linux", en el cual se describe cómo defender el sistema operativo frente a las objeciones que se le hacen desde diferentes ámbitos, fundamentalmente por parte de compañías de software.

SERVICIOS AL LECTOR

Dentro de los servicios se incluye una guia de sitios web relacionados con la temática, un glosario de términos utilizados habitualmente por los usuarios de GNU/Linux, y abundante información sobre bibliografia recomendada.

En esta sección se describen, además, las diferentes fuentes de información sobre el sistema operativo, fundamentalmente aquellas que están en idioma español.

A lo largo del libro encontrarán estos recuadros con información complementaria:

CURIOSIDADES

Datos divertidos y locuras varias que resultan necesarios para ser un experto animador de reuniones sociales.

DEFINICIONES

Después de leer estas definiciones, no existirán más palabras incomprensibles ni temas que lo inhiban.

IDEAS

Trucos para realizar distintas tareas de manera más rápida y efectiva. Consejos sabrosos para todo conocedor del tema.

ATENCIÓN

Problemas típicos o errores frecuentes con los que se cruza el usuario inquieto, y los secretos para evitarlos.

DATOS UTILES

Información valiosa, datos precisos y actualizados, sitios web clave y respuestas a las preguntas frecuentes.

SOLO PARA GENIOS

Información y trucos para usuarios avanzados. ¡Todos llevamos un genio dentro (el asunto es saber encontrarlo)!

NOVEDAD

Comentarios sabrosos acerca de las novedades incluidas en la última versión y las mejoras logradas en sus aplicaciones.

ON WEB

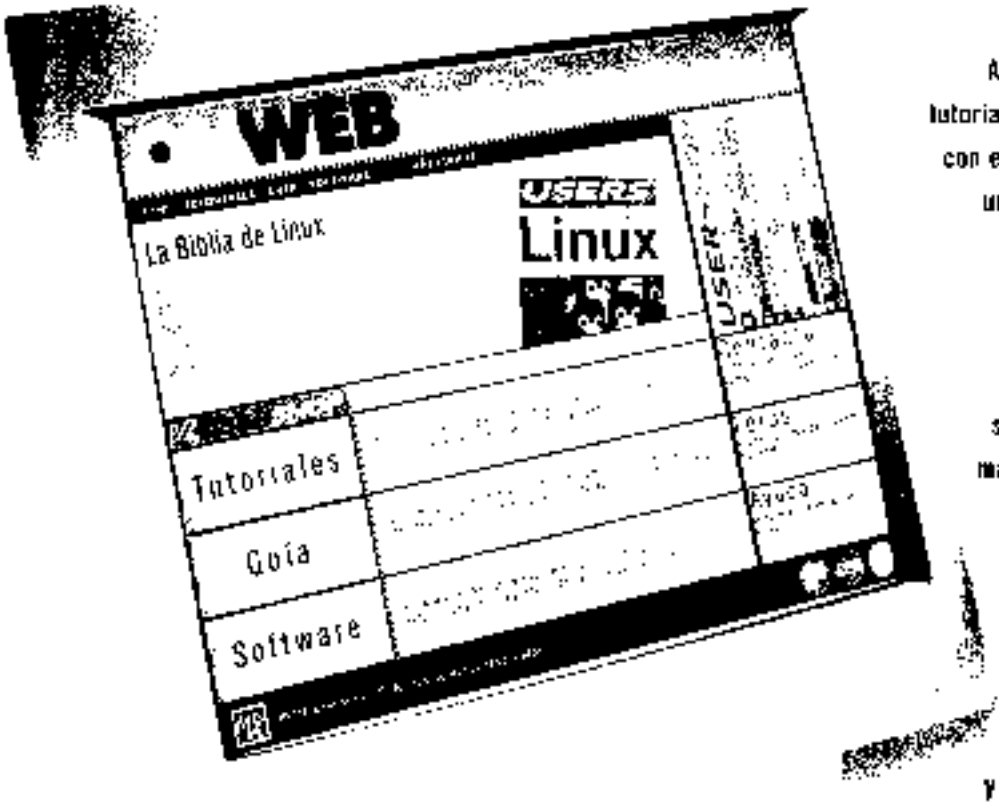
Información, recursos, software o ejemplos del libro que están publicados en el sitio web exclusivo para lectores: onweb.tectimes.com.

UNA NUEVA DIMENSIÓN EN LIBROS

SOFTWARE RECURSOS

• WEB

ONWEB.TECTIMES.COM

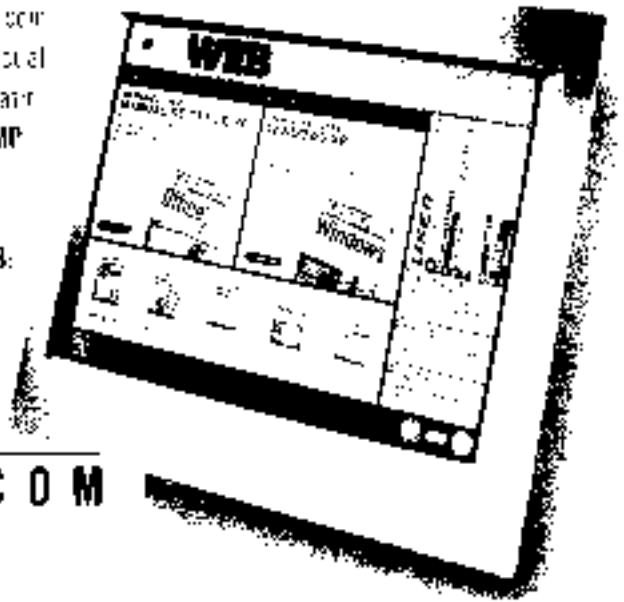


Aquí encontrará diferentes tutoriales en video relacionados con el libro. Sólo deberá hacer un clic en Ver Tutorial para bajar el video a su PC.

Una completa guía con sitios web, para acceder a más información y recursos útiles que le permitirán profundizar sus conocimientos.

Las mejores aplicaciones y herramientas accesorias, ejemplos y listados del libro para que no tenga que invertir su tiempo en transcribirlos.

OnWeb, el sitio que le permitirá aprovechar al máximo cada uno de nuestros libros con contenido exclusivo: la mejor selección de software y los ejemplos analizados en el texto, tutoriales en video y una completa guía de sitios de Internet. Además, un foro a través del cual podrá realizar interconsultas con otros lectores y usuarios, debatir con ellos y estar en contacto con la editorial. Como siempre, **MP Ediciones**, a la vanguardia en la divulgación de la tecnología.

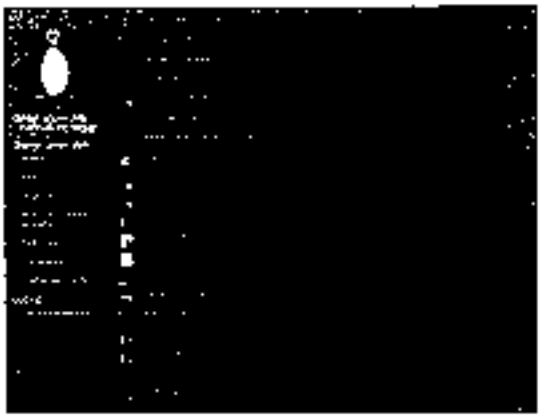


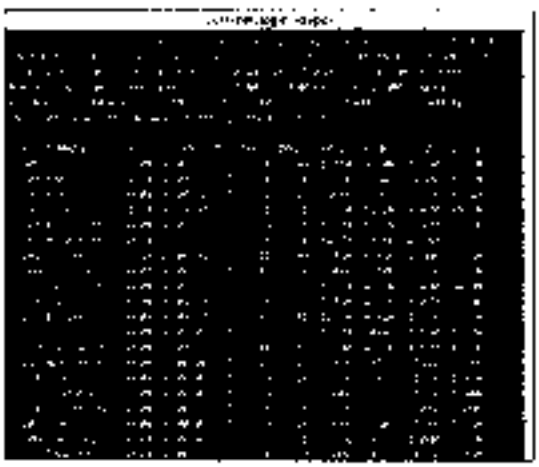
BIENVENIDO A LOS SERVICIOS EXCLUSIVOS DE ONWEB:

Ingrese al sitio onweb.tectimes.com. La primera vez que acceda, deberá registrarse con un nombre de usuario y una clave. Para completar el proceso de registro, se le hará una pregunta referida al libro y se le solicitarán sus datos personales.

ONWEB.TECTIMES.COM

Contenido



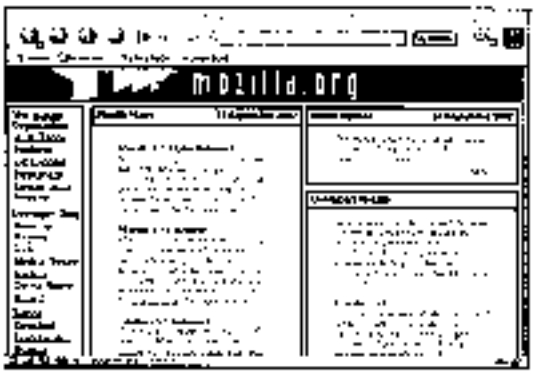


CAPÍTULO 6

El modo gráfico

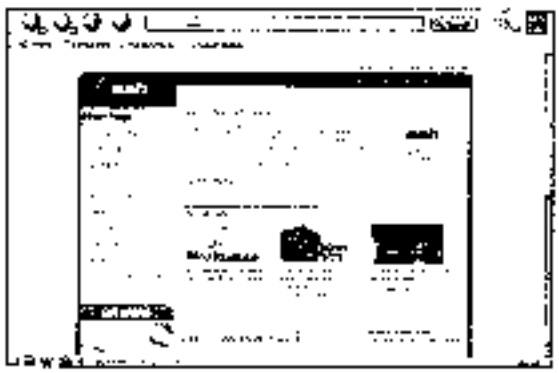
CAPÍTULO 7

GNU/Linux en red



CAPÍTULO 8

Login remoto



CAPÍTULO 9

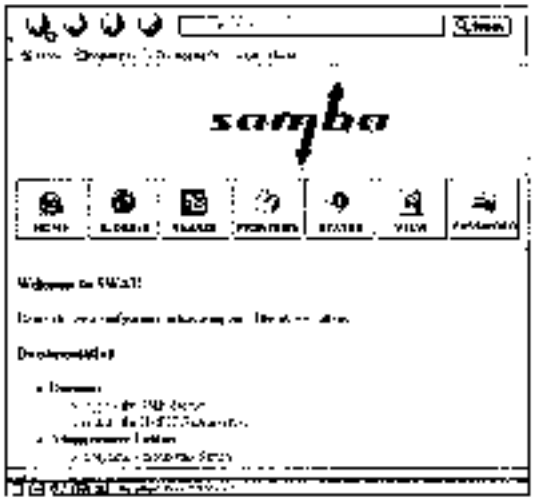
Instalar un servidor web

CAPÍTULO 10

El servicio NFS

CAPÍTULO 11

Interconexión en red con Windows



CAPÍTULO 12

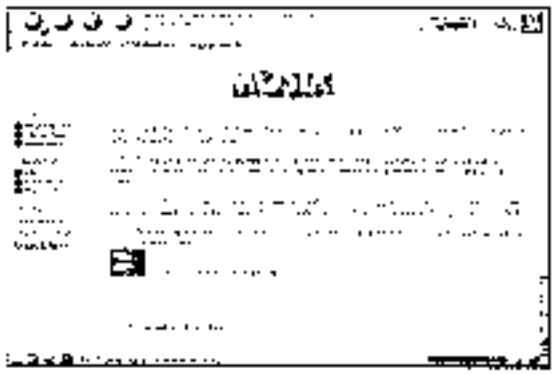
Firewalls y proxies

CAPÍTULO 13

El sistema VNC

CAPÍTULO 14

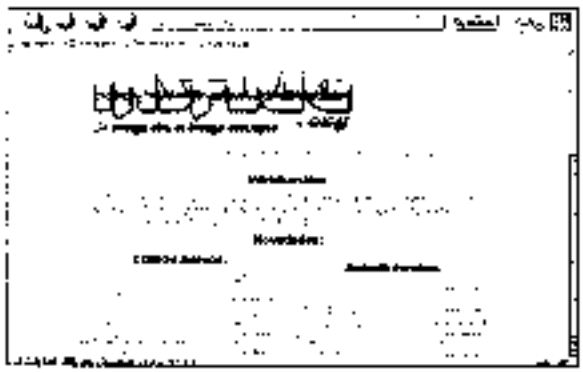
Clusters Beowulf



APÉNDICES

Documentos oficiales

SERVICIOS AL LECTOR



Introducción

¡Bienvenidos! Ésta es la primera palabra que se merece toda persona que se interesa en aprender sobre el sistema operativo GNU/Linux. En el mundo de GNU/Linux, encontrarán muchos conceptos interesantes, cosas complicadas, cosas divertidas (otras aburridas); pero, sobre todo, hallarán algo nuevo que es totalmente diferente de lo que venían usando. Ahora bien, ¿no será un problema el hecho de que este sistema sea tan nuevo?, ¿costará demasiado implementar con GNU/Linux las cosas que hacíamos en otros sistemas operativos? La respuesta a estas dos preguntas es muy sencilla: no, si sabemos cómo hacerlas.
GNU/Linux posee conceptos basados en UNIX, el sistema operativo padre de casi todos los sistemas que vemos hoy. Por lo tanto, si ya sabían montar una red en Windows NT o programar en lenguaje C bajo DOS, la cosa no será tan distinta.
La diferencia con los demás sistemas operativos radica en la filosofía de desarrollo y distribución de este producto del trabajo de miles de personas de todo el mundo.
Por lo tanto, no se asusten ni se preocupen. Con un poco de paciencia –y la ayuda del libro que en este momento tienen en sus manos– lograrán convertirse en experimentados usuarios del sistema operativo GNU/Linux en muy poco tiempo.

A quién y cómo

Este libro está dirigido a todo aquel que quiera relacionarse (o ya tenga relación) con el sistema operativo GNU/Linux. Se tratan los más diversos temas, de forma que quien no conoce absolutamente nada acerca de este sistema operativo puede comenzar a dar sus primeros pasos con él, y quien ya tiene experiencia puede encontrar conceptos útiles y formas de desarrollar soluciones a los diferentes problemas que se presentan en los sistemas informáticos.
Tal como La Biblia original, este libro puede ser leído de varias formas. Si ustedes no conocen absolutamente nada de GNU/Linux y quieren comenzar a dar sus primeros pasos, entonces lo más recomendable será comenzar por el primer capítulo, y leerlos todos, uno por uno, hasta llegar al **Capítulo 4**. A partir de allí, se tratan algunos conceptos avanzados que suponen cierta experiencia por parte del lector. Por esto, una vez que lleguen a este capítulo, dejen descansar el libro unos días (una semana o dos) y dediquen ese tiempo a obtener experiencia haciendo uso de todos los recursos que aprendieron en los capítulos anteriores.
Si son usuarios experimentados, entonces lo mejor será que vayan directamente a lo que necesitan. Los primeros capítulos son introductorios y sirven para explicar la instalación y el uso básico del sistema operativo. A partir del **Capítulo 3** encontrarán toda la información necesaria para manejar el sistema operativo a fondo, principalmente desde el modo texto. En los últimos capítulos hallarán información para usuarios avanzados, así como direcciones para obtener más recursos en Internet.

Génesis

Este capítulo, sumamente teórico, tiene como objetivo principal aclarar todos los conceptos filosóficos que giran alrededor de la palabra GNU/Linux.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

¿Qué es el Software Libre?

El Software Libre (nótense las mayúsculas) es un concepto que no es nuevo. La idea principal detrás de estas palabras es la libertad de compartir la información. Actualmente, existen varias modalidades de desarrollo y distribución de software, que son:

- **Software propietario:** estos programas y aplicaciones suelen estar desarrollados por empresas que licencian el código fuente del programa y no permiten su redistribución. Cuando una persona adquiere un sistema propietario, generalmente sólo obtiene una versión precompilada de ese código fuente, con permiso para ser usada en "n" cantidad de computadoras. Si se lo quiere utilizar en un número mayor de computadoras, se debe pagar por cada licencia un precio fijado por la empresa. Además, al no obtener el código fuente, el usuario está imposibilitado de modificar el programa o ver cómo funciona internamente. De más está decir que es ilegal copiar un programa de este tipo a un amigo o a un familiar.

- **Shareware:** ésta es una modalidad de desarrollo y distribución que tuvo mucho éxito a finales de los '80 y en los '90. La idea detrás del shareware es la de "probar antes de comprar". Generalmente, estos programas no dejan de ser propietarios, pero se distribuye una versión reducida (o con límite de uso temporal) por los BBS y CDs de revistas para que la gente pueda probarlos. Si le gusta, puede pagar un precio por la versión completa, la cual tampoco incluye el código fuente (generalmente). La versión *shareware* es de libre distribución, la versión completa, no.

***Figura 1**. Los desarrolladores de shareware poseen su propia asociación, la Asociación de Profesionales del Shareware (**www.asp-shareware.org**).*

- **Freeware:** en conjunto con el shareware, se desarrolló el freeware. Aquí las cosas son un poco mejores para el usuario final, ya que un programa que es freeware es un programa que está completo, y puede ser utilizado y distribuido libremente. El problema es que estos programas generalmente no tienen la calidad de uno propietario y de Software Libre. En los programas freeware tampoco se incluye el código fuente, por lo que éstos sufren también las limitaciones mencionadas en las otras modalidades.

- **Software Libre:** y, finalmente, llegamos al Software Libre. El punto máximo de libertad tanto para el desarrollador como para el usuario. Para que un programa sea Software Libre, debe cumplir con cuatro requisitos básicos. El primero de ellos es que el programa pueda ser utilizado sin ningún tipo de limitación. El segundo requisito es que pueda ser distribuido libremente y copiado a cuantas computadoras sea necesario. El tercero es muy sencillo: el programa siempre debe estar acompañado del código fuente (o de una carta al usuario en donde se ofrezca un acceso a él). Este requisito es muy importante, ya que al disponer del código fuente, los usuarios pueden hacerle modificaciones y, así, adecuarlo mejor a sus necesidades. El último punto, en realidad, no es un requisito: un programa que es Software Libre **se puede vender**. Incluso se puede vender una versión modificada de un programa de Software Libre. Siempre y cuando se respeten los nombres de los autores originales y los tres puntos anteriores, no hay ninguna restricción para hacer algo de dinero con un programa de Software Libre.

***Figura 2**. En el sitio oficial del proyecto GNU (**www.gnu.org**) encontrarán muchísima documentación relativa al Software Libre.*

Ahora bien, como ustedes saben, los programas se rigen por licencias de uso y distribución. La licencia más utilizada por el Software Libre es la Licencia Pública General

(*GPL*, ver **APÉNDICE A**), la cual se encarga de proteger los derechos del autor y de los usuarios. Cuando se distribuye un programa de Software Libre bajo la GPL, siempre se incluye un archivo con el texto de la licencia. La Licencia Pública General está desarrollada y soportada por la Fundación del Software Libre.

¿Qué es GNU/Linux?

Muchos querrán saltear esta sección alegando que no estarían leyendo este libro si no supieran lo que es GNU/Linux. En realidad, esta sección es más importante de lo que parece.

GNU/Linux es el primer sistema operativo basado en UNIX que es 100% Software Libre. Si bien anteriormente había otros sistemas operativos de libre distribución (como MINIX), éstos no eran totalmente Software Libre, ya que eran regidos por licencias más restrictivas.

GNU/Linux es un proyecto que ya lleva 20 años en desarrollo, y lo estará por muchos más, ya que se asienta sobre una base de cientos de programadores de todas partes del mundo. Muchas veces me preguntan si no es posible que "el que hace Linux un día se vuelva rebelde y quiera hacer que su sistema sea propietario". La respuesta es, obviamente, negativa. No existe una persona "que hace Linux". GNU/Linux es un conjunto de componentes desarrollados por muchas personas que trabajan en muchos proyectos. No es un único paquete (aunque muchos de ustedes lo hayan instalado como tal). Es prácticamente imposible parar un proyecto de estas magnitudes.

Hablando técnicamente, GNU/Linux es un sistema operativo de software libre basado en UNIX, que cumple las normas POSIX. Su base es un núcleo monolítico llamado Linux (a secas), desarrollado originalmente por Linus B. Torvalds a principios de la década de los noventa. Su estructura general es la típica de cualquier sistema UNIX (*núcleo – intérprete de comandos – aplicaciones*), aunque actualmente debe de ser el más desarrollado de ellos. Cuenta con una interfaz gráfica llamada Xfree86 (versión libre del sistema de ventanas Xwindow original del MIT) y con muchas aplicaciones para realizar las más diversas tareas, desde procesamiento de textos hasta montaje de servidores de red, pasando por aplicaciones multimedia y juegos.

LA FREE SOFTWARE FOUNDATION

La FSF (Fundación del Software Libre) es una fundación sin fines de lucro, que se mantiene con las donaciones que hacen los usuarios y empresas de forma directa. Posee su sitio oficial en Internet (www.gnu.org). Allí, encontrarán muchísima documentación acerca del sistema operativo GNU/Linux y todos los proyectos que están relacionados con él.

Historia de GNU/Linux

A principios de la década del ‘80, un científico del área de inteligencia artificial del Instituto de Tecnología de Massachussets (MIT), llamado Richard Stallman (RMS, para los amigos), decidió comenzar a desarrollar un sistema operativo libre, ya que en esa época la única opción que tenían los usuarios de computadoras era adquirir un software propietario. El nombre de este sistema es GNU, acrónimo de las palabras “¡GNU No es Unix!”. La idea principal era que el sistema mantuviera un grado de similitud con el ya conocido UNIX sin compartir una sola línea de código fuente. Idea que, obviamente, fue hecha realidad en algunos años.

Para finales de los ‘80, el sistema estaba prácticamente completo. Disponía de editores de texto, de intérpretes de comandos, de compiladores, de debuggers, etc. Sólo faltaba un componente: el núcleo (kernel).

Casualmente, en el otro lado del mundo (Finlandia), un estudiante llamado Linus B. Torvalds desarrolló un núcleo compatible con UNIX, sin saber que iba a formar parte del sistema operativo más revolucionario del mundo de la informática. Así fue como apareció Linux, el núcleo del sistema GNU. Y es así como hoy tenemos un sistema completo a nuestra disposición, el sistema GNU/Linux.

Paralelamente con el desarrollo de este sistema operativo, surgió la Fundación del Software Libre, la cual fomenta, entre otras cosas, la utilización de herramientas de Software Libre en las computadoras de todo el mundo.

GNU es uno de los proyectos más grandes de Software Libre. En él participan miles de personas de todo el mundo colaborando con el desarrollo de aplicaciones para el sistema, documentación, mantenimiento del servidor web y muchísimo más.

Cuando hablamos de Libertad, en el mundo del Software Libre lo hacemos en el sentido más filosófico de la palabra. Hablamos de la libertad de tener un programa completo (incluido su código fuente), de la libertad de usarlo, copiarlo, modificarlo, venderlo, de la libertad de compartirlo con los otros. Ése es el espíritu del sistema GNU/Linux.

El software libre presenta una innumerable cantidad de ventajas para el desarrollador frente a otros sistemas desarrollados bajo modelos cerrados. La primera y principal ventaja es que el desarrollador obtendrá ayuda de parte de personas que quizá ni siquiera conoce, gracias a la gran Red de redes.

La segunda ventaja es que su proyecto crecerá mucho más rápido que antes gracias a la cantidad de colaboradores que quieran sumarse a la causa (siempre que ésta sea buena).

Ahora bien: ¿cómo se mantiene un proyecto basado en la colaboración? No es muy difícil. Para mantener el proyecto de desarrollo funcionando se necesitan principalmente dos cosas: una buena organización y capital. Vayamos por partes.

Recordemos que los proyectos de Software Libre generalmente se basan en la participación de miles de personas de alrededor del mundo que poseen una sola cosa en

común: el espíritu de colaboración. Después de este nivel, casi no hay estructura jerárquica. Este tipo de proyectos posee un líder, que es el encargado de seleccionar el material proveniente de las colaboraciones y mantener el proyecto a flote. Para esto, los líderes deben tener una personalidad muy especial. Deben saber decir "no" cuando algo no sirve y deben saber encaminar el proyecto en el rumbo adecuado.
Existen muchas organizaciones con grandes estructuras que comenzaron como pequeños proyectos de Software Libre. Ejemplos de ellas son GNOME, KDE, Samba, etc. Cuando el proyecto se inicia, la necesidad de invertir capital en él es prácticamente nula. Los proyectos de Software Libre comienzan como un pasatiempo para un desarrollador y, si van por el buen camino, terminan convirtiéndose en proyectos masivos que mueven grandes cantidades de capital.
Pero ¿de dónde proviene el capital si la gente no pagará por el programa? Ésa es la pregunta que mucha gente se hace sin conocer a fondo el sistema del Software Libre. Como mencionamos al principio, el Software Libre nos da la libertad de comprarlo y venderlo. ¡Vender un software es parte de la libertad! Ahora bien, cuando se vende un producto de Software Libre, no se debe limitar al cliente dándole un binario ejecutable y un contrato que le impida copiarlo a sus amigos. Si es Software Libre, debe ir provisto del código fuente y se le debe dar la posibilidad al cliente de copiarlo sin problemas, haciendo uso de alguna de las licencias de software que hoy disponemos (GPL es una de ellas).
Otra forma de obtener capital para invertir en un proyecto de Software Libre es distribuyendo el software en un medio físico y otorgando servicios extra para los que lo adquieran. Un buen ejemplo de esto son las distribuciones de GNU/Linux que actualmente existen en el mercado. Si aún no tienen en claro qué es una distribución, podemos hacer una analogía con una empresa productora de automóviles. Un auto está compuesto por muchas partes; las empresas automotrices seleccionan los mejores componentes y los ensamblan. Entonces, cuando uno compra un auto, confía en la empresa automotriz que mejor haya ensamblado esos componentes. Con las distribuciones ocurre exactamente lo mismo: el sistema operativo GNU/Linux está compuesto por muchos componentes (programas, librerías y documentación). Cuando uno compra una distribución, adquiere un paquete armado por una empresa que se ha tomado el trabajo de seleccionar los mejores paquetes para un requerimiento dado.
Retomando, tomemos como ejemplo la distribución Mandrake, que es una de las más famosas en estos tiempos. Mandrake es una distribución que se vende en varias modalidades: una versión estándar y una deluxe son las más conocidas. Además, Mandrake es una distribución que puede ser descargada de Internet sin ningún tipo de limitación. Entonces, ¿cuál es el beneficio de adquirir alguna de las versiones empaquetadas de Mandrake? La respuesta es simple: cuando se adquieren estas distribuciones, se obtienen los CDs principales de la distribución, CDs extra con miles de aplicaciones, manuales impresos y, lo más importante de todo, el servicio técnico al usuario.
Este último servicio sólo es provisto a los usuarios que adquieran cualquiera de las

versiones empaquetadas de Mandrake, y representa una ventaja para el usuario final, ya que, cuando tenga un problema, podrá llamar al servicio técnico, y éste intentará orientarlo en el buen camino para que solucione su problema.
Otra modalidad de conseguir capital para solventar un proyecto de Software Libre son las inversiones de las grandes empresas. Existen compañías como IBM, Hewlett-Packard y Sun Systems que apoyan muy fuertemente los desarrollos de Software Libre realizando inversiones millonarias en los proyectos. ¿Por qué hacen esto? Porque saben que las herramientas son realmente de buena calidad.

La definición de Software Libre

(Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001 Free Software Foundation).

Traducción: equipo de traductores a español de GNU.

Mantenemos esta definición de Software Libre para mostrar claramente qué debe cumplir un programa de software concreto para que se lo considere Software Libre.
El "Software Libre" es un asunto de libertad, no de precio. Para entender el concepto, debes pensar en "libre" como en "libertad de expresión", no como en "barra libre" [N. del T.: en inglés, una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión].
"Software Libre" se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es Software Libre si los usuarios tienen todas estas libertades. Así, pues, deberías tener la libertad de distribuir copias, sea con modificaciones o sin ellas, sea gratis o cobrando una cantidad por la distribución, a cualquiera y en cualquier lugar. El hecho de ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.
También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de utilizarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica.
La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero debes tener la libertad de distribuir estos formatos si encontraras o desarrollaras la manera de crearlos.
Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, debes tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el Software Libre.
Para que estas libertades sean reales, deben ser irrevocables mientras no hagas nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no le hayas dado motivos, el software no es libre.
Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir Software Libre , mientras no entren en conflicto con las libertades centrales. Por ejemplo, copyleft ("izquierdo de copia", expresado muy simplemente) es la regla que implica que, cuando se redistribuya el programa, no se podrán agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.
Así, pues, quizás hayas pagado para obtener copias de software GNU, o tal vez las hayas obtenido sin ningún coste. Pero independientemente de cómo hayas conseguido tus copias, siempre tienes la libertad de copiar y modificar el software, e incluso de vender copias.
"Software Libre" no significa "no comercial". Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del Software Libre ha dejado de ser inusual; el software comercial libre es muy importante.
Es aceptable que haya reglas acerca de cómo empaquetar una versión modificada, siempre que no bloqueen, como consecuencia de ello, tu libertad de publicar versiones modificadas. Reglas como "Si haces disponible el programa de esta manera,

EL NOMBRE DEL SISTEMA

Ahora que saben que el sistema completo se llama Proyecto GNU y que Linux es sólo el núcleo de éste, ¿no les parece una injusticia que todo el mundo lo llame simplemente Linux a secas? Lamentablemente, ese concepto erróneo ya está demasiado asentado en la gente, aunque siempre tendrán oportunidad de explicarle a alguien de dónde proviene realmente el nombre del sistema.

debes hacerlo disponible también de esta otra" pueden ser igualmente aceptables, bajo la misma condición. (Observa que una regla así todavía te deja decidir si publicar o no el programa). También es aceptable que la licencia requiera que, si has distribuido una versión modificada y el desarrollador anterior te pide una copia de ella, debas enviársela.

En el proyecto GNU, utilizamos "copyleft" para proteger de modo legal estas libertades para todos. Pero el Software Libre sin "copyleft" también existe. Creemos que hay razones importantes por las que es mejor usar "copyleft", pero si tus programas son Software Libre sin ser copyleft, podemos utilizarlos de todos modos.

Visita la página Categorías de Software Libre (18.000 caracteres, **www.gnu.org/philosophy/categories.es.html**) para ver una descripción de las diferencias que hay entre el "Software Libre", el "software con copyleft" ("izquierdo de copia"), y cómo otras categorías de software se relacionan unas con otras.

A veces, las normas de control de exportación del Gobierno y las sanciones mercantiles pueden restringir tu libertad de distribuir copias de los programas a escala internacional. Los desarrolladores de software no tienen el poder de eliminar o sobrepasar estas restricciones, pero lo que pueden y deben hacer es rehusar el hecho de imponerlas como condiciones de uso del programa. De esta manera, las restricciones no afectarán a actividades y gente fuera de las jurisdicciones de estos gobiernos.

Cuando se habla de Software Libre, es mejor evitar términos como "regalar" o "gratis", porque esas palabras implican que lo importante es el precio, y no la libertad. Algunos términos comunes tales como "piratería" conllevan opiniones que esperamos no apoyes. Visita la página "Palabras y frases confusas que vale la pena evitar" (**www.gnu.org/philosophy/words-to-avoid.es.html**), donde encontrarás una discusión acerca de estos términos. También tenemos una lista de traducciones de "Software Libre" en varios idiomas.

Por último, fíjate en que los criterios establecidos en esta definición de Software Libre requieren que los analicemos cuidadosamente para interpretarlos. Para decidir si una licencia de software concreta es una licencia de Software Libre, lo juzgamos basándonos en estos criterios para determinar si tanto su espíritu como su letra en particular los cumplen. Si una licencia incluye restricciones contrarias a nuestra ética, la rechazamos, aun cuando no hayamos previsto el problema en estos criterios. A veces, un requisito de una licencia plantea una situación que necesita de una reflexión minuciosa, e incluso conversaciones con un abogado, antes de que podamos

MÁS INFORMACIÓN SOBRE RMS

Richard Stallman (RMS) es una persona muy abierta y siempre dispuesta a responder las dudas de todos los entusiastas de la filosofía del Software Libre. Pueden encontrar su página personal en www.stallman.org.

decidir si la exigencia es aceptable. Cuando llegamos a una conclusión, a veces actualizamos estos criterios para que sea más fácil ver por qué ciertas licencias se pueden calificar o no como de Software Libre.
Si te interesa saber si una licencia concreta es de Software Libre, mira nuestra lista de licencias (**www.gnu.org/philosophy/license-list.es.html**). Si la licencia que te preocupa no está en la lista, puedes preguntarnos enviándonos un correo electrónico a **<licensing@gnu.org>**.

¿Qué es el Open Source?

El Open Source es una organización que se encarga de fomentar el uso de **sistemas de código abierto**, concepto significativamente diferente del de Software Libre. Un sistema de código abierto no necesariamente tiene que estar regido bajo la Licencia Pública General, sino que puede estar regido por alguna de las tantas licencias de la Iniciativa Open Source (**www.opensource.org**). Uno de los fundadores de esta iniciativa es Eric Raymond, famoso hacker del mundo de GNU/Linux.
Obviamente, aunque siempre hay discusiones entre los fanáticos del Open Source y del Free Software, la verdad siempre la tiene el desarrollador, que es quien decide usar alguna de estas dos modalidades para el desarrollo de su programa, según cuáles sean sus requerimientos.

***Figura 3**. Si no les queda bien definida la diferencia entre Software Libre y software de código abierto, el sitio **www.opensource.org** puede ser un lugar de ayuda.*

En resumen…

El movimiento del Software Libre es el principal responsable de esta gran revolución que significa GNU/Linux. La evolución en la tecnología de las comunicaciones (Internet, por ejemplo) ha permitido que millones de personas de todo el mundo unan sus esfuerzos para trabajar en una causa común: el desarrollo de toda una plataforma de Software Libre para cubrir el más amplio espectro de necesidades. De esta forma, los usuarios de computadoras de todo el mundo tendrán una alternativa libre para elegir, a la hora de solucionar sus problemas mediante la informática.

GNU/Linux - La solución

Seguramente, ustedes habrán oído numerosas veces hablar acerca de este sistema operativo (sí, el del pingüino). Y seguramente se habrán preguntado cuál es la razón de su éxito. La puedo resumir en dos palabras: **la libertad**. GNU/Linux es un sistema operativo desarrollado y distribuido de forma totalmente libre. No hay empresas que definan los cursos por seguir en cuanto a su desarrollo; tampoco hay intereses de por medio que eviten la evolución del sistema operativo: todo es totalmente transparente, y si cualquier persona tiene dudas acerca de esta transparencia, no tiene más que navegar por los sitios de los diferentes proyectos que conforman GNU/Linux para obtener todo el código fuente y la documentación de ese proyecto de forma totalmente gratuita.

Todo usuario es libre de elegir el tipo de software que quiere usar para resolver sus problemas, pero día a día podemos ver cómo las herramientas de Software Libre son las que están comenzando a acaparar cada vez más sectores del mercado del software. ¿Necesitan ejemplos? Mozilla como navegador web, StarOffice como suite de oficina, Evolution como sistema de correo electrónico y organizador personal, XMMS como reproductor de archivos MP3, y la lista podría seguir mucho más. Todos esos programas tienen un costo monetario igual a cero y pueden ser descargados, usados y distribuidos con total libertad.

Actualmente, el pingüino ocupa el segundo puesto en el ranking mundial de los sistemas operativos más utilizados (obviamente, el primer puesto lo ocupa el archiconocido sistema propietario de las ventanas). Y aunque muchos pregonen lo contrario, todo parece indicar que se convertirá en el sistema operativo estándar de las computadoras de escritorio en los próximos años. Y esto tiene una causa principal: la comunicación. Internet ha sido el principal causante de toda esta revolución llamada Software Libre. Cuando una empresa cualquiera desarrolla un software propietario que es útil para mucha gente e impone muchas restricciones para su uso y

su distribución, no pasan semanas hasta que aparece un proyecto de Software Libre con el objetivo de reemplazarlo y hacer al bien común. Estas versiones de Software Libre generalmente se tornan mucho más funcionales que las versiones desarrolladas por las empresas. Entonces, los usuarios inteligentes optan por usar esta versión de Software Libre, y los otros se quedan en el camino, pagando precios mucho más altos y obteniendo menos beneficios.

Lo mismo pasa con los gobiernos del mundo: países como Alemania, Italia y México ya están instalando GNU/Linux en sus sistemas informáticos gubernamentales, con lo que obtienen grandes ahorros de dinero que luego puede ser invertido en otras áreas de igual o mayor importancia.

Ahora bien, si toda esta tecnología libre está desarrollada y funciona muy bien, ¿por qué no es utilizada en los ámbitos en los que el costo del software puede perjudicar notablemente a la economía general? Para ser más precisos, ¿por qué no se utiliza Software Libre en las escuelas, municipios y otros organismos gubernamentales de nuestro país? Si hay que recortar gastos, reemplazar el software propietario por Software Libre es una excelente idea. Es de conocimiento público que muchos organismos estatales y privados están informatizados y usan menos licencias de las que deberían tener. Entonces, ¿qué sale más barato? ¿Pagar todas las licencias que faltan o instalar Software Libre? La respuesta es obvia.

Quiero aclarar que no estoy en contra de la comercialización del software propietario; tampoco estoy en contra de las grandes corporaciones, pero cuando un país está en crisis, lo primero que debe hacer es buscar la forma de beneficiar a su pueblo, a su economía local. Si se instala Software Libre, el dinero se queda en la economía local, en lugar de ir a parar a las arcas de grandes corporaciones extranjeras. Cuando se implementa Software Libre, sólo hay que invertir una mínima cantidad de dinero en servicios (instalación, mantenimiento, administración, etc.), que puede ser perfectamente manejado por empresas y profesionales locales. La implementación de este tipo de software ayuda a que los gobiernos obtengan la independencia económica que, en mayor o menor medida, siempre es necesaria.

Todo cambio es traumático. Una transición de este tipo debe implementarse por etapas, haciendo pruebas piloto para poder tener sobre la mesa todas las situaciones de impedimento que puedan llegar a aparecer, pero al final el resultado siempre es beneficioso.

En estos tiempos en que es necesario reducir gastos y reactivar el mercado tecnológico local, el Software Libre es **la única solución**. Por eso hago un llamado a la reflexión a las autoridades de los diferentes organismos públicos y me pongo a disposición de cualquier autoridad que desee obtener más información.

LA BIBLIA DE

Capítulo 2

Primeros pasos

Con la ayuda de este capítulo daremos nuestros primeros pasos en el uso de GNU/Linux. Experimentados, ¡abstenerse!

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

El proceso de inicio

Como siempre, vamos a comenzar por el principio: analizaremos el proceso de inicio de un sistema informático convencional. Téngase en cuenta que los conceptos explicados en esta sección del libro son sólo aplicables a la arquitectura de computadoras x86, originalmente de Intel.

Fase 1: Encendido de la computadora

Cuando se enciende la computadora, una serie de tareas se realizan de forma totalmente automática. La primera de ellas es la carga del BIOS (*Basic Input Output System*). Este sistema ofrece un conjunto básico de instrucciones para trabajar con el hardware a muy bajo nivel. Luego de que se carga el BIOS y se realiza una revisión general del sistema (memoria, dispositivos de almacenamiento, teclado, etc.), la computadora ya está lista para cargar un sistema operativo.

Fase 2: El gestor de arranque

La segunda fase consiste en cargar un gestor de arranque. Éste es un programa muy pequeño que entra en la MBR (Master Boot Record) de un disco rígido, y su función principal es actuar como cargador del núcleo del sistema operativo. Todo sistema operativo digno de llamarse como tal debe utilizar un gestor de arranque (aunque muchas veces esto es totalmente transparente para el usuario).

En GNU/Linux hay dos gestores de arranque muy famosos: el primero de ellos es LILO, un gestor que dominaba casi todas las distribuciones. LILO es muy fácil de configurar y es sumamente práctico si se lo utiliza correctamente. El segundo es GRUB, un gestor un poco más avanzado (permite configurar un inicio mucho más personalizado) pero que, a la vez, resulta más complejo de configurar para muchos usuarios. GRUB comenzó a ganar terreno cuando se descubrió un pequeño bug en LILO que imposibilitaba instalar el cargador en discos rígidos mayores a los 10 GB. Obviamente, este error fue corregido al poco tiempo.

Ambos gestores de arranque permiten la configuración de múltiples sistemas operativos, por lo que en una misma computadora podemos tener instalados GNU/Linux y Windows o cualquier otro sistema operativo (incluso, dos distribuciones de GNU/Linux). Más adelante se verá en detalle cómo configurar el gestor de arranque LILO. GRUB es usado en muy pocas distribuciones actualmente, dada la dificultad que presenta para configurarlo y administrarlo. Muchas distribuciones dan la opción de elegir entre un gestor y otro en el momento de la instalación. Yo les recomiendo usar LILO.

Fase 3: Carga del núcleo

Llegó la hora de cargar el núcleo y comenzar con los preparativos para que el sistema operativo entre en la etapa "utilizable". Una vez que LILO (o GRUB) ha cargado el núcleo Linux, éste comienza a instalar en memoria los controladores necesarios para utilizar los dispositivos más básicos, como pantalla, teclado, unidades de almacenamiento, reloj del sistema, etc. En esta etapa, se dice que el sistema está en "modo núcleo". Los más experimentados entenderán que este modo es el "estado real" de la computadora, en el que todos los programas y controladores aún pueden acceder al hardware sin ningún tipo de restricción.

Además, en esta etapa también se cargan los módulos del sistema. Los módulos son los controladores de dispositivos específicos como placas de sonido y placas de red. Estos módulos deben ser configurados por el usuario (no se preocupen, cuando se instala una distribución, el sistema de instalación detecta la mayor parte de los dispositivos y decide qué módulos instalar).

Una vez que la carga de los módulos termina, el sistema entra en modo protegido (en GNU/Linux se llama "modo usuario" o *user mode*). En este modo, los programas no pueden acceder directamente al hardware del sistema, sino que tienen que hacerlo a través de funciones especiales de programación (llamadas al sistema, etc.) provistas por el núcleo.

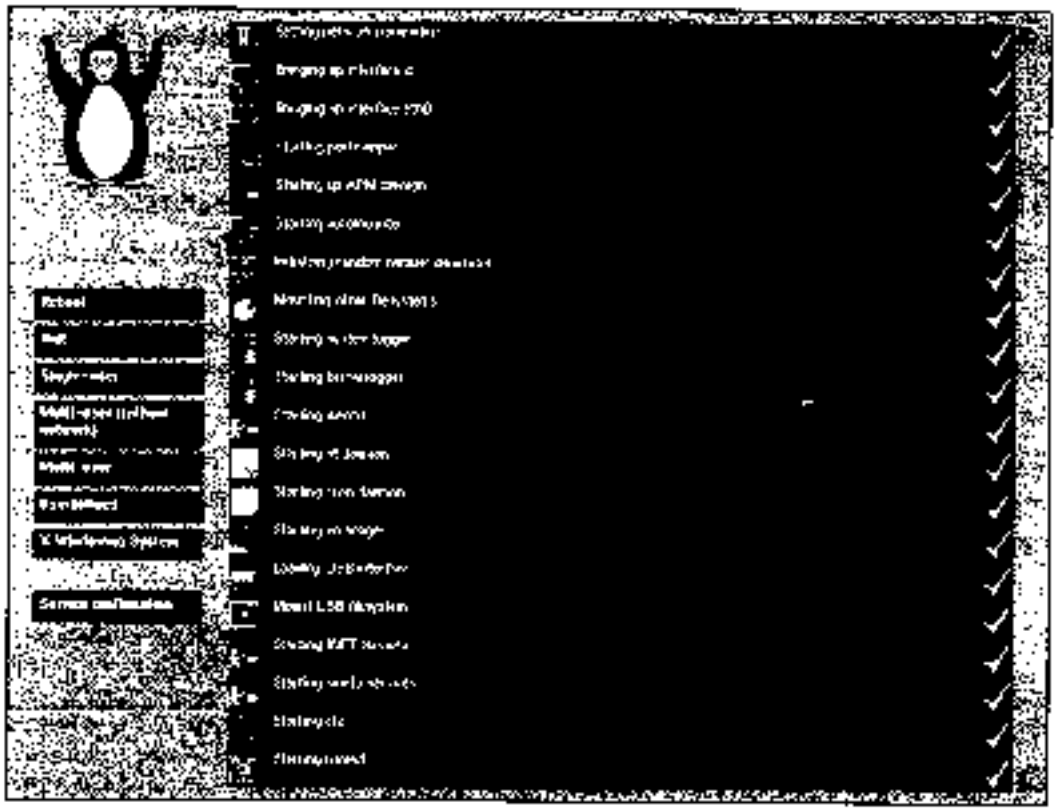


Figura 1. Algunas distribuciones (Mandrake, por ejemplo) presentan la información del proceso de inicio en forma gráfica; otras (como Slackware) sólo lo hacen en modo texto.

Fase 4: Programas de usuario

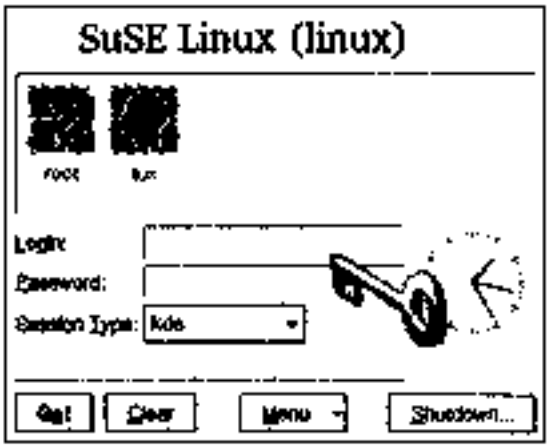


Figura 2. KDM es uno de los tantos gestores de sesiones para el modo gráfico de GNU/Linux.

El sistema ya está listo para comenzar a cargar en memoria los programas del usuario. El primer programa se llama **gestor de sesiones**. Éste, básicamente, nos permite registrarnos en el sistema con un nombre de usuario y una contraseña. Recordemos que GNU/Linux es un sistema operativo multiusuario, por lo que muchas personas pueden emplear una misma computadora, cada una con su perfil propio. Por ahora sólo diremos que el gestor de sesiones se ejecuta automáticamente y es el primer programa con el que el usuario deberá interactuar, ingresando su nombre de usuario y su contraseña.

Existe un usuario llamado **root** (raíz), que es el que actúa como administrador de la computadora. Su función es mantener el orden, y posee privilegios que le permiten manejar cualquier archivo o recurso de la computadora, aun cuando éste sea propiedad de otro usuario. El usuario **root** existe en todas las instalaciones de GNU/Linux, y su clave de acceso es definida en el momento de la instalación de la distribución.

El sistema de archivos

Existen muchas definiciones acerca de lo que es un sistema de archivos; nosotros, para simplificar, diremos que se trata de la forma en que el sistema operativo estructura los datos en la unidad de almacenamiento.
En GNU/Linux, los datos se ordenan en archivos y directorios (como en tantos otros sistemas operativos). La diferencia especial radica en que (generalmente) los programas no son almacenados cada uno en su propio directorio, sino que sus diferentes componentes

EL SISTEMA EXT2

El sistema de archivos de GNU/Linux más utilizado ha sido, durante mucho tiempo, el EXT2 (Extended 2). Éste es un sistema que permite la atribución de usuarios y permisos a los archivos y directorios.

LOS NUEVOS SISTEMAS

En la actualidad, el EXT2 está quedando en desuso, y es reemplazado por sistemas más avanzados, como EXT3 o ReiserFS. Muchas distribuciones ya dan la opción de elegir alguno de estos sistemas de archivos avanzados.

están dispersos por todo el sistema de archivos. Así, contamos con un directorio específico para todos los ejecutables, otro para la documentación, otro para las librerías, etc. La siguiente tabla les ayudará a reconocer la inmensa cantidad de directorios que componen el árbol de GNU/Linux.

DIRECTORIO	DESCRIPCIÓN
/	Directorio raíz, desde aquí "cuelgan" todos los demás directorios del sistema.
/bin	En este directorio se almacenan archivos binarios ejecutables. Generalmente, encontraremos los archivos correspondientes a los comandos básicos del sistema.
/boot	En este directorio se almacena el núcleo Linux, así como los archivos de configuración necesarios para su uso.
/dev	Éste es un directorio muy especial. Los archivos que están aquí dentro representan los diferentes dispositivos del sistema. Más adelante trataremos este tema en detalle.
/etc	Aquí se almacenan todos los archivos de configuración de GNU/Linux y de los demás programas de usuario.
/home	Este directorio contiene los subdirectorios personales de los usuarios del sistema. Cada usuario posee su propio directorio, en el que puede almacenar archivos personales, tales como documentos, programas y archivos de configuración.
/lib	En este directorio se almacenan las librerías de programación básicas de GNU/Linux.
/mnt	Este directorio contiene subdirectorios que actúan como puntos de montaje. Desde aquí podemos acceder al contenido de otras particiones o unidades.
/root	Éste es el directorio personal del usuario root.
/sbin	Aquí se encuentran los archivos binarios ejecutables correspondientes a los comandos de administración del sistema.
/proc	Este directorio almacena archivos que contienen información sobre el sistema (CPU, memoria, dispositivos PCI, Plug'n Play, etc.)
/usr	En este directorio encontraremos "todo lo demás": archivos de documentación, programas, más librerías, código fuente, etc.
/var	En este directorio se almacenan varias cosas, como la cola de impresión, los archivos de registración (log), etc. Será interesante dar un vistazo a este directorio.

Como pueden observar, el directorio raíz de un sistema GNU/Linux contiene muchos subdirectorios, todos imprescindibles para el correcto funcionamiento del sistema. Pero (¿lamentablemente?) el tema no termina acá: el directorio /usr, tal como mencionamos anteriormente, es un extenso apartado del sistema de archivos que contiene muchos archivos útiles. Por esta razón, tendremos que analizarlo en detalle:

DIRECTORIO	DESCRIPCIÓN
/usr/X11R6	Éste es el subdirectorio principal del sistema de ventanas XfreeB6. Aquí encontrarán los archivos binarios correspondientes al entorno gráfico, así como algunos archivos de documentación, configuración y librerías de programación.
/usr/bin	En este directorio se almacenan los archivos binarios ejecutables correspondientes a los programas de usuario que estén instalados en el sistema.
/usr/doc	Este directorio es utilizado para almacenar la documentación de los programas instalados. Aquí encontrarán muchos subdirectorios (con nombres generalmente descriptivos), que contienen archivos de texto que pueden ser visualizados con cualquier editor. En algunas distribuciones, la documentación se almacena en **/usr/share/doc**.
/usr/games	Si bien este directorio no se usa mucho, antiguamente era el lugar indicado para almacenar todos los juegos.
/usr/include	Aquí se encuentran los archivos de cabecera para la programación en lenguaje C.
/usr/lib	En este directorio se almacenan las librerías de programación utilizadas por los programas de usuario.
/usr/sbin	En este directorio encontrarán más aplicaciones también relacionadas con la administración del sistema.
/usr/src	Este directorio sirve para guardar el código fuente de los programas. Entre otros, encontraremos el código fuente del núcleo Linux.

El sistema de archivos es mucho más extenso, y presentar todos los subdirectorios nos llevaría varios tomos de una enciclopedia. Será trabajo del lector comenzar a indagar por los diferentes rincones de la partición para ver con qué tesoro se puede encontrar.

Moverse por los directorios

Ya instalamos nuestro sistema GNU/Linux y describimos las diferentes fases por las que pasa el proceso de inicio. Hemos llegado al punto en el que el sistema nos pregunta por un usuario y una contraseña (los cuales fueron configurados durante la instalación del sistema operativo). Ahora bien, según qué distribución se use, las cosas pueden ser presentadas de diferentes formas para el usuario. Concretamente, el tema es el siguiente: el proceso de registro puede ser mostrado al usuario de dos formas: gráfica o textual. El sistema operativo GNU/Linux puede ser operado en todo momento

en cualquiera de estos dos modos. En el modo texto, el usuario sólo ve caracteres en la pantalla y puede ejecutar programas mediante el tipeo de comandos en una consola. Los programas no son gráficos, sino que solamente despliegan letras blancas sobre un fondo negro.
Puede parecer aburrido, pero este modo es mucho más rápido para algunas operaciones que el modo gráfico.
En el modo gráfico, el usuario controla el sistema operativo mediante un lindo sistema de ventanas, haciendo uso del mouse frente a una pantalla que despliega infinidad de colores. Ahora que tenemos en claro cómo son estos dos sistemas, volvamos al proceso de inicio.
Si el proceso de inicio se ejecuta en modo texto (sólo verán una pantalla negra con una palabra: **login:**), entonces sólo tendrán que limitarse a ingresar el usuario (root, por ejemplo) y la contraseña, y ya pueden comenzar a tipear comandos del sistema.
Pero si el proceso de registro aparece en modo gráfico (una ventana con el logo de nuestra distribución, generalmente), entonces tendrán que escribir el usuario y la clave en una caja de diálogo, y el sistema automáticamente los llevará al entorno gráfico. Aquí no tendrán que tipear ningún tipo de comando.
Nosotros vamos a comenzar trabajando en el modo texto, por lo que si a ustedes, lectores, les aparece un inicio en modo gráfico, presionen la combinación de teclas **CTRL+ALT+F1** para pasar a una consola en modo texto y registrarse en el sistema como usuarios de tal. Si en algún momenton quiere, volver al modo gráfico, sólo presionen **ALT+F7**.
Bueno, ya estamos en una sesión de modo texto, por lo que podemos comenzar a tipear nuestros primeros comandos. Lo primero que haremos será pedir al sistema que nos despliegue un listado de directorios. Si hemos entrado como usuario root, el listado será algo similar al siguiente:

```
[root@localhost /root]# ls

Desktop     Pictures          evolution  gimp.txt
imacplayer-1.0-1.0.src.rpm Documents  directorios.txt  gimp.abw   imacplayer-
1.0-1.0.ppo.rpm  mujeres125.jpg
[root@localhost /root]#
```

UBICACIÓN DE LOS ARCHIVOS

En este libro presentamos la ubicación estándar de los archivos en la mayoría de las distribuciones de GNU/Linux. Luego, el lector puede encontrarse con diferencias en distribuciones muy particulares.

Como pueden observar, al ejecutar el comando ls, el sistema nos devuelve en pantalla un listado de todos los archivos y directorios contenidos en el directorio en el que nos encontramos actualmente (/root, en el caso del ejemplo). Si queremos un listado un poco más informativo y ordenado, sólo tendremos que usar el parámetro -l. El resultado será el siguiente:

```
[root@localhost /root]# ls -l
total 480
-rw-r--r-- 1 root root 771      Feb    20 14:29 #directorios.txt#
drwx------ 3 root root 4096     Dec    4 15:24 Desktop
drwx------ 2 root root 4096     Jan    2 2001 Documents
drwxr-xr-x 3 root root 4096     Dec    28 20:10 Pictures
-rw-r--r-- 1 root root 289      Feb    20 14:27 directorios.txt
drwx------ 6 root root 4096     Feb    5 15:49 evolution
-rw-r--r-- 1 root root 3362     Jan    2  2001 gimp.abw
-rw-r--r-- 1 root root 4350     Jan    6 11:20 gimp.txt
[root@localhost /root]#
```

La primera columna nos indica cuáles son los permisos de uso de cada archivo/directorio. En próximos capítulos analizaremos el tema de permisos en detalle y haremos un uso intensivo de ellos. Por ahora, lo único que tienen que saber es que si la primera letra es una d, entonces se trata de un directorio; en caso contrario, es un archivo común. Las demás columnas nos indican datos varios, como el dueño del archivo, el tamaño de éste, la fecha de creación y el nombre. Ahora vamos a "salir" de este directorio, haciendo uso del típico comando cd del siguiente modo:

```
cd /
```

El símbolo del sistema se convertirá en algo parecido al siguiente:

```
[root@localhost /]#
```

EL COMANDO Y LOS PARÁMETROS

Es usual que los que vienen de sistemas como Windows o DOS intenten usar el comando 'cd..' o 'cd/'. Esto es incorrecto. En GNU/Linux hay que separar bien el comando de su parámetro. El uso adecuado sería 'cd ..' (un espacio en el medio) y 'cd /'.

En este momento nos encontramos en el directorio raiz de la partición de GNU/Linux, por lo que si pedimos un listado de directorios, nos encontraremos con los directorios que anteriormente analizamos. Para entrar en un directorio, sólo hay que usar el comando cd del siguiente modo:

```
cd [directorio]
```

Entonces, el siguiente ejemplo es perfectamente válido:

```
cd /usr
```

Ahora que ya saben cómo moverse por los directorios, pueden comenzar a hacer sus propias investigaciones por todo el árbol de directorios de GNU/Linux. Si bien muchos nombres de archivos no son muy descriptivos, es importante que empiecen a acostumbrarse visualmente a la estructura de directorios en los sistemas UNIX.

Cerrando el sistema

Vamos a cerrar el sistema. Cuando se usa GNU/Linux, apagar la computadora sin cerrar el sistema significa un gran potencial de pérdida de datos. En modo texto, cerraremos el sistema con el siguiente comando:

```
halt
```

El sistema comenzará a cerrar todos los recursos, y cuando todo esté listo, nos lo informará en pantalla y podremos apagar la computadora.
Cabe destacar que sólo el usuario root (el administrador) puede cerrar el sistema.

EL MODO TEXTO

Llamamos "modo texto" a aquel que sólo nos permite manejarnos con caracteres textuales en la pantalla. Las aplicaciones no mostrarán gráficos de ningún tipo, y a lo sumo podremos apreciar diferentes colores de caracteres.

OTRAS FORMAS DE CERRAR

También pueden cerrar el sistema utilizando el comando **reboot** (para reiniciar la computadora) o **shutdown -h now** (que cumple la misma función que **halt**).

En resumen...

En este segundo capítulo hemos visto los detalles más importantes que un usuario debe saber luego de instalar su sistema GNU/Linux. No todo el mundo está acostumbrado a trabajar con sistemas UNIX, y algunos conceptos pueden asustar a más de uno, aunque sean tanto o más sencillos de entender que sus equivalentes en otros sistemas operativos. Mediante el uso de los comandos de listado y el cambio de directorios, hemos analizado, uno por uno, los directorios más importantes del sistema (si, son muchos, pero con el tiempo y la práctica, lograrán recordarlos). Además, aprendimos algo muy importante: cómo apagar el sistema. Mucha gente no sabe cómo hacerlo, y luego de instalar su flamante distribución, apaga la computadora sin cerrar la sesión y se encuentra con que la próxima vez que inicia la computadora, ésta comienza a hacer una revisión del sistema de archivos en busca de errores. Esto es porque en el proceso de apagado se ejecutan ciertos procesos de descarga de unidades y programas, y si estos procesos no son ejecutados, se corre el peligro de perder datos.
Suficiente por ahora. En el próximo capítulo, nos dedicaremos a aprender los comandos más importantes para convertirnos en expertos administradores de GNU/Linux en modo texto. Relajen sus dedos y prepárense para teclear.

LA BIBLIA DE

Capítulo 3

Trabajando en el modo texto

En este capítulo aprenderán los conceptos básicos para manejar un sistema GNU/Linux desde el entorno textual. Olvídense del mouse por un rato, y ¡a teclear!

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

Manejo de archivos

Comenzaremos trabajando sobre los archivos de nuestro sistema. Tengan en cuenta que trabajaremos en el sistema como usuario administrador (**root**), ya que aún no sabemos cómo crear un usuario nuevo. Por esta razón, presten mucha atención a los archivos con los que están trabajando, ya que si dañan o borran algún archivo importante del sistema, las pérdidas pueden ser incalculables (a nivel software, claro). De todas formas, intentaré guiarlos lo mejor posible desde estas líneas. Recuerden que trabajaremos exclusivamente en el modo texto.

Copiando archivos

Una de las funciones más típicas es la de copiar archivos. ¿Quién no necesita alguna vez copiar un archivo de un directorio a otro? Bueno, desde la línea de comandos, tenemos un comando que nos permite realizar esta función. El comando se llama **cp** (a diferencia de otros sistemas como MSDOS, en donde se llama **copy**) y tiene la siguiente estructura básica:

```
cp [opciones] [archivo] [destino]
```

Los parámetros **[archivo]** y **[destino]** pueden contener rutas completas a directorios. El parámetro **[opciones]** puede ser obviado, aunque más adelante veremos algunas opciones importantes.
Ahora veamos cómo usar este comando de forma básica. El siguiente es un ejemplo válido:

```
cp /etc/lilo.conf /tmp
```

El comando del ejemplo anterior se encargará de copiar el archivo **lilo.conf** (ubicado en el directorio **/etc**) al directorio **/tmp**.
Los experimentados en MSDOS se estarán preguntando si pueden usar comodines (*wildcards*). ¡Claro que sí! Y éstos funcionan exactamente igual que en todos los demás sistemas operativos. Veamos los siguientes ejemplos:

```
cp *.png /graficos
cp archivo*.* /documentos
```

El primer ejemplo se encargará de copiar todos los archivos que posean la extensión **.png** en el directorio gráfico. El segundo ejemplo copiará todos los archivos que comiencen con la palabra "archivo" y finalicen con cualquier extensión, en el directorio *documentos*. Como mencionamos anteriormente, el comando **cp** permite utilizar algunas opciones para personalizar aún más su funcionamiento. Veamos las más importantes:

OPCIÓN	DESCRIPCIÓN
-a	Preserva todos los atributos de archivo.
-b	Realiza una copia de seguridad antes de copiar.
-i	Pide confirmación antes de sobrescribir archivos.
-R	Copia los archivos recursivamente en todos los subdirectorios.
-u	Copia un archivo sólo cuando éste es más nuevo que el que se encuentra actualmente en el destino
-v	Muestra información detallada mientras se realiza la operación.
help	Muestra información sobre su utilización en pantalla.

Borrar archivos

Otra de las acciones que serán necesarias en más de una ocasión es la de borrado de archivos. En GNU/Linux existe un comando que nos permite realizar esta tarea rápidamente. Su forma es la siguiente:

```
rm [opciones] [archivo]
```

El comando **rm** se limita a eliminar del sistema el archivo definido como parámetro. De esta forma, para eliminar el archivo que copiamos en el ejemplo anterior, sólo tenemos que ejecutar el siguiente comando:

```
rm /tmp/lilo.conf
```

El comando **rm** también admite comodines. Las opciones del comando **rm** son las siguientes:

OPCIONES	DESCRIPCIÓN
-f	Elimina archivos sin pedir confirmación
-i	Pide confirmación antes de eliminar un archivo.
-r	Elimina recursivamente todos los archivos en todos los subdirectorios. También borra los directorios.
-v	Muestra en pantalla el nombre de cada archivo que va siendo eliminado.

Ahora bien, hemos visto cómo eliminar archivos, pero no, cómo eliminar directorios. En definitiva, no es muy diferente: para borrar un directorio simplemente tienen que usar el comando **rmdir** del siguiente modo:

```
rmdir [directorio]
```

El sistema se encargará automáticamente de eliminar el directorio definido como parámetro. Recuerden que para eliminar un directorio, siempre es necesario que éste se encuentre totalmente vacio (sin archivos).

Creación de directorios

Crear directorios bajo GNU/Linux desde la linea de comandos es un proceso realmente sencillo. Para esto, el sistema nos provee de una herramienta llamada **mkdir**. Su forma es la siguiente:

```
mkdir [directorio]
```

De esta manera, el comando **mkdir temporal** creará un directorio llamado **temporal**, al cual podremos acceder con el comando **cd**, tal como lo vimos anteriormente.

Permisos de acceso

Una de las caracteristicas de los sistemas operativos UNIX es la posibilidad de que cada archivo tenga ciertos permisos de acceso y utilización. En realidad, esta característica la provee el sistema de archivos.
Cada archivo de nuestro sistema de archivos (incluidos los directorios) posee ciertos permisos de acceso. Éstos son, básicamente, tres: Lectura (**r**), Escritura (**w**) y Ejecución (**x**). Estos permisos pueden ser aplicados al usuario dueño del archivo, al grupo de usuarios al cual pertenece y a todos los demás usuarios del sistema. Esto significa que podemos asignar permisos de ejecución al dueño de un archivo, pero podemos asignar permisos de sólo lectura a todos los demás usuarios.
Para ir entendiendo un poco mejor las cosas, vamos a teclear algunos comandos. Existe un parámetro del comando **ls** que nos permite visualizar información detallada de los archivos y directorios. Uno de los datos que nos ofrece son los permisos de cada archivo y de cada directorio. El parámetro es **-l**, y ofrecerá un listado similar al siguiente:

```
[localhost:~/temporal] hfarena% ls -l
-rw-r--r--  1 hfarena  staff  25 May 13 15:28 documento.txt
[localhost:~/temporal] hfarena%
```


Como podemos ver, en el directorio **temporal** sólo hay un archivo llamado **documento.txt**. La columna que más nos interesa ahora es la primera (**-rw-r--r--**). En ésta, cada letra indica el permiso asignado. Si en lugar de una letra aparece un símbolo menos ("-"), entonces quiere decir que el permiso no está asignado. Ahora bien, ¿cuál es la estructura de permisos? Muy sencilla: el primer carácter puede contener una letra **d** si es un directorio, o un simbolo "-" si es un archivo. Los demás caracteres los dividiremos en grupos de tres: los primeros tres son los permisos asignados al usuario, el segundo trio son los permisos asignados al grupo, y los últimos tres caracteres son los permisos asignados a todos los demás usuarios. Veamos algunos ejemplos:

drwxrwxrwx Sería un directorio (primer carácter con letra "d") que poseeria permisos de lectura, de escritura y de ejecución para su dueño, para su grupo y para todos los demás usuarios del sistema.

-rwx------ Sería un archivo (primer carácter con símbolo "-") que sólo podría ser leído, modificado y ejecutado por su dueño.

-r--r--r-- Sería un archivo que podria ser leido por todos los usuarios, pero no podría ser modificado ni ejecutado por ninguno.

-rw-rw-r- Sería un archivo que podria ser leido por todos los usuarios del sistema, pero sólo podria ser modificado por su propietario y por su grupo.

Es recomendable ejecutar un **ls -l** en diferentes directorios para ver otros ejemplos de permisos.
Suficiente teoria. Vayamos a la práctica (muchos lectores se estarán preguntando ya cómo modificar los permisos de un archivo). Para manejar permisos, GNU/Linux nos provee de una herramienta llamada **chmod**. Su estructura es la siguiente:

```
chmod  [quien] [+/-] [permiso]  [archivo]
```

El parámetro **[quien]** puede contener una letra **u** (usuario dueño del archivo), una letra **g** (grupo) o una letra **o** (todos los demás). El segundo parámetro nos permite asignar un permiso con el carácter "+" o denegar un permiso con el carácter "-". Por último, se definen los permisos en si y el archivo al cual serán asignados. Veamos algunos ejemplos:

```
chmod u+x documento.txt
```

Asigna permiso de ejecución del archivo **documento.txt** sólo para su dueño.

```
chmod ug+r documento.txt
```

Asigna permisos de lectura al archivo **documento.txt** tanto para su dueño como para el grupo al que pertenece.

```
chmod o-x script.pl
```

Deniega la ejecución del archivo **script.pl** a todos los usuarios que no pertenezcan al grupo del usuario propietario de ese archivo.
Una última aclaración: el permiso de ejecución sólo debe ser asignado a archivos binarios ejecutables y archivos de script (interpretados por lenguajes de programación como Perl). En caso de que se asigne un permiso de ejecución a un archivo que no fue concebido originalmente para ser ejecutado, los resultados en el comportamiento del sistema pueden ser inesperados.

Dueños

Otra propiedad que poseen los archivos en el sistema GNU/Linux es que tienen propietarios. Esto quiere decir que cada archivo le corresponde a un usuario o a un grupo de usuarios.
El sistema es muy sencillo de comprender: si creamos un archivo con cualquier aplicación (un documento en un editor de textos, por ejemplo), ese archivo tendrá registrado como dueño nuestro nombre de usuario. Si queremos que ese archivo sea propiedad de otro usuario, tendremos que usar el comando **chown**. La forma básica de este comando es:

```
chown usuario [archivo(s)]
```

De esta forma, el nuevo propietario del archivo podrá realizar modificaciones a sus permisos, ya que es el nuevo dueño.

Enlaces

Los enlaces son accesos directos a un archivo, desde otra zona del sistema de archivos. Por ejemplo, si queremos crear un acceso directo al archivo **/etc/passwd** en el directorio **/root**, sólo tendremos que usar el siguiente comando:

```
ln /etc/passwd /root/mienlace
```

El comando **ln** se encargará automáticamente de crear el enlace. El primer parámetro indica hacia dónde apunta ese enlace, y el segundo parámetro es el nombre del enlace. De esta forma, cuando realicemos cualquier operación sobre **/root/mienlace**, ésta, en realidad, será efectuada en **/etc/passwd**. Obviamente, el enlace es sólo un apuntador al archivo, por lo que de ninguna manera tendrá el mismo tamaño que el archivo apuntado.

El proceso de montaje

Para acceder a otras particiones y dispositivos, GNU/Linux (al igual que muchos otros sistemas UNIX) utiliza el proceso de montaje. Este sistema se basa en la siguiente metodología de uso:

1. Si el dispositivo es removible (un disco floppy o un CD), el usuario lo ingresa en la lectora.
2. Mediante el comando **mount**, asigna ese dispositivo a un directorio denominado punto de montaje.
3. Desde el punto de montaje, el usuario podrá leer y escribir los datos almacenados en ese dispositivo.
4. Una vez que haya terminado de usar el dispositivo, el usuario debe eliminar el enlace entre el dispositivo y el punto de montaje haciendo uso del comando **umount**.

Veamos un ejemplo para aclarar las cosas. Si un usuario quiere utilizar un disco floppy, entonces debe insertarlo en la disquetera y ejecutar el siguiente comando:

```
mount /dev/floppy /mnt/floppy
```

El comando **mount** requiere dos parámetros. El primero de ellos es el dispositivo por montar. Recuerden que en el directorio **/dev/** están todos los archivos que representan los diferentes dispositivos básicos de nuestro sistema. Por consiguiente, **/dev/floppy** representa nuestra primera disquetera. El segundo parámetro que requiere el

comando **mount** es el punto de montaje, o sea: el directorio que usaremos para acceder a ese disco flexible. El directorio **/mnt/floppy** que pusimos en el ejemplo puede no existir. En tal caso, el usuario puede crear su propio directorio o utilizar cualquier otro (aunque la primera opción es la más recomendable).
Veamos otro dispositivo removible: un CD. Para montarlo, el usuario deberá tipear la siguiente línea:

```
mount /dev/cdrom /mnt/cdrom
```

La explicación anterior es totalmente aplicable a este nuevo ejemplo. Ahora bien, si queremos montar otras unidades no removibles (como particiones), el usuario deberá tener en cuenta lo siguiente: los discos rígidos IDE se enuncian bajo los archivos **/dev/hda** (para el primer IDE master) y **/dev/hdb** (para el IDE esclavo; generalmente éste es la unidad de CD-ROM). Entonces, si queremos montar la primera partición de nuestro IDE maestro, deberemos tipear:

```
mount /dev/hda1 /mnt
```

El archivo **/dev/hda1** representa la primera partición. Si quisiéramos acceder a la segunda partición, sería **/dev/hda2**, y así sucesivamente.
Ahora bien, una vez que hayan terminado de utilizar el dispositivo, deben desmontarlo. Es muy importante que, si se trata de un dispositivo removible, no lo quiten de la lectora antes de desmontar. En tal caso, corren riesgo de pérdida de datos. Para desmontar un dispositivo (sea removible o no) se utiliza el comando **umount** del siguiente modo:

```
umount  [punto_de_montaje]
```

Por ejemplo, para desmontar una unidad de floppy que fue montada en **/mnt/floppy**:

```
umount /mnt/floppy
```

¿QUÉ ES EL SISTEMA DE ARCHIVOS?

El sistema de archivos es la forma que utiliza el sistema operativo para estructurar el almacenamiento de datos. En MSDOS, el sistema se llama FAT. En GNU/Linux, el sistema de archivos más difundido se llama EXT2 (Extended 2). Aunque actualmente hay otros sistemas más avanzados que están comenzando a implementarse lentamente, como el ReiserFS.

Variables de entorno

Muchas veces, los programas que se ejecutan en el sistema (concurrentemente o no) necesitan compartir información. Una de las maneras de llevar esto a cabo es mediante las variables de entorno.
Definiremos variable de entorno como una posición en memoria identificada con un nombre y en la cual se guarda algún tipo de dato.
La definición es muy similar a la de las variables globales en la programación: son variables accesibles por todas las funciones. Éstas son algunas variables de entorno que encontrarán en la mayoría de las distribuciones de GNU/Linux:

VARIABLE	DESCRIPCIÓN
DISPLAY	El nombre del dispositivo en el que el servidor de ventanas X mostrará su salida.
HOME	El directorio home.
LOGNAME	El nombre utilizado en el login.
PATH	Una cadena que contiene el listado de directorios en donde se buscarán los programas si no son encontrados en el directorio actual.
PS1	Define el tipo de línea de comandos.
SHELL	El shell estándar. En nuestro caso, SHELL=/bin/bash.
TERM	Esta variable define el tipo de terminal que estamos utilizando.

Para definir una variable de entorno, utilizamos el comando **export** con el siguiente formato:

```
export NOMBRE=valor
```

De esta manera, para definir una variable con el nombre "LISTA" y valor 17, ejecutamos el siguiente mandato:

```
export LISTA=17
```

Para el caso de la sugerencia (agregar un directorio a la variable **PATH**), sólo hay que utilizar el siguiente formato:

```
export PATH="$PATH:/directorio"
```

Con este comando, hemos agregado la cadena **:/directorio** a la variable **PATH**.

Pueden utilizar las variables de entorno en cualquier parte de su línea de comandos. Por ejemplo:

```
cd $HOME
```

Cambiará automáticamente a su directorio personal.

Manejo de procesos

Ahora que sabemos cuáles son los comandos básicos para manejarnos en el modo textual del sistema operativo GNU/Linux, comenzaremos a realizar algunas tareas de administración avanzada. Una de ellas, y quizá la más importante, es el manejo de procesos.

¿Qué son los procesos?

Para que los lectores puedan comprenderlo fácilmente, un proceso es un programa que se está ejecutando en un momento dado en el sistema. Cuando usamos normalmente el sistema operativo GNU/Linux, existe una serie de procesos ejecutándose constantemente sin que el usuario logre darse cuenta de ello, y que hacen que el sistema sea utilizable.
El primer proceso que se ejecuta se llama **init**, y es el que se encarga de establecer el nivel de ejecución correspondiente (ya hablaremos de los niveles de ejecución más adelante). Luego, se ejecutan los procesos correspondientes a las terminales (**tty's**), los sistemas de **login** (el programa que pregunta por un usuario y una contraseña) y, finalmente, el intérprete de archivos (**bash**, generalmente). Todos esos procesos están funcionando en el sistema cuando ustedes están trabajando en él.
Los procesos se manejan de forma jerárquica: cada proceso es lanzado desde un proceso padre. Este proceso lanzado se llama proceso **hijo**. Por eso, en GNU/Linux, todos los procesos son hijos del gran padre **init**. Si cerramos el proceso del intérprete de comandos (**bash**), automáticamente estaremos cerrando todos sus procesos hijos, generalmente los programas que el usuario esté ejecutando.
Hay, básicamente, dos tipos de procesos: los de usuario y los *demonios*. Los procesos de usuario son los programas que el usuario utiliza generalmente, y que están conectados a la terminal, esto es: el programa aparece en pantalla y el usuario interactúa con él. Los demonios son procesos que no están conectados a la terminal. Funcionan solos, no requieren la interacción del usuario, y en general se usan para

programas que funcionan constantemente como servidores de red, programas administrativos, etc.
Por último, diremos que en la jerga del sistema UNIX hay varias palabras que el usuario debe conocer. Los procesos pueden estar **vivos**, o pueden estar **muertos**, son **lanzados** (nacen) o pueden ser **matados**. También pueden ser **zombies**, o pueden quedar **huérfanos** (sin proceso padre). Y, como mencionamos anteriormente, son **padres e hijos** de otros procesos.
Ahora si pongamos manos a la obra y comencemos a trabajar con el sistema de procesos de GNU/Linux.

Listado de procesos

Para obtener un listado de procesos en cualquier sistema UNIX, disponemos de un comando llamado **ps**. Su forma original es:

```
ps
```

Automáticamente aparecerá en pantalla un listado similar al siguiente:

```
localhost% ps
PID     TT      STAT    TIME        COMMAND
268     std     Ss      0:00.29     -tcsh (tcsh)
291     std     R       0:00.04     sh
localhost%
```

Esta tabla nos muestra varios datos útiles. En la primera columna (PID), nos indica el número ID del proceso en cuestión. Cada proceso posee un número que lo identifica, el cual es generado aleatoriamente y nos será de utilidad cada vez que queramos hacer referencia a ese proceso. La segunda columna nos indica la terminal en la cual está funcionando ese proceso. La tercera columna muestra el estado del proceso; la cuarta columna, la cantidad de tiempo que el proceso ha consumido para ser lanzado y, por último, la columna **COMMAND** señala qué comando ha sido utilizado para ejecutar dicho proceso.
Ahora bien, en el ejemplo del libro sólo vemos dos procesos en el listado. Esto, obviamente, no es real. Lo que sucede es que **ps**, si se lo utiliza sin parámetros, sólo muestra los procesos de usuario. No, los demonios. Si queremos un listado real y completo de todos los procesos del sistema, deberemos usar el comando **ps ax**. Aparecerá en pantalla algo mucho más extenso:

```
localhost% ps ax
PID    TT      STAT    TIME        COMMAND
1      ??      SLs     0:00.02     /sbin/init
2      ??      SL      0:00.79     /sbin/mach_init
21     ??      Ss      0:00.38     kextd
70     ??      Ss      1:15.54     CoreServices/WindowServer
72     ??      Ss      0:00.30     update
103    ??      Ss      0:00.36     /sbin/autodiskmount -va
126    ??      Ss      0:00.71     configd
152    ??      Ss      0:00.07     syslogd
158    ??      Ss      0:00.01     /usr/libexec/CrashReporter
180    ??      Ss      0:00.32     netinfod -s local
187    ??      Ss      0:00.27     lookupd
200    ??      Ss      0:00.45     CoreServices/coreservicesd
207    ??      Ss      0:00.02     inetd
268    std     Ss      0:00.29     -tcsh (tcsh)
291    std     R       0:00.04     ah
localhost%
```

Como se puede apreciar, la mayor parte de los procesos posee un simbolo "??" en la columna **TT**. Esto se debe a que son procesos demonios, que no están siendo ejecutados en ninguna terminal específica.

Eliminando procesos en ejecución

Para eliminar (*matar*) un proceso en ejecución necesitamos, básicamente, dos cosas: utilizar el comando **kill** con el número PID correspondiente al proceso que queremos matar. Será el primer paso, entonces, pedir un listado de procesos al sistema con el comando **ps** para averiguar cuál es su número PID correspondiente. Luego, utilizaremos el comando **kill** del siguiente modo:

```
kill [PID]
```

Esto le enviará al proceso una señal 15 (terminación por software), y el programa se cerrará automáticamente. Ahora bien, existen algunos procesos que ignoran por completo a la señal 15, y no se cerrarán con el comando **kill** a secas. A estos procesos retobados, vamos a eliminarlos enviándoles una señal 9 (señal de eliminación "a toda costa"). Para esto simplemente usaremos el comando **kill** de la siguiente forma:

```
kill -9 [PID]
```

Ahora hemos eliminado ese programa que no se quería cerrar. Tengan mucho cuidado cuando realicen pruebas con el comando **kill**, ya que pueden eliminar procesos de vital importancia para el sistema, y luego, los resultados en el funcionamiento pueden ser totalmente inesperados.

Top: una herramienta de utilidad

Disponemos de una herramienta muy útil para visualizar el estado del sistema, con un listado de procesos en tiempo real. Esta utilidad se llama **top** y está incorporada en casi todas las versiones de UNIX, incluido GNU/Linux. Para ejecutarla, simplemente tendremos que tipear el comando **top** en la terminal.

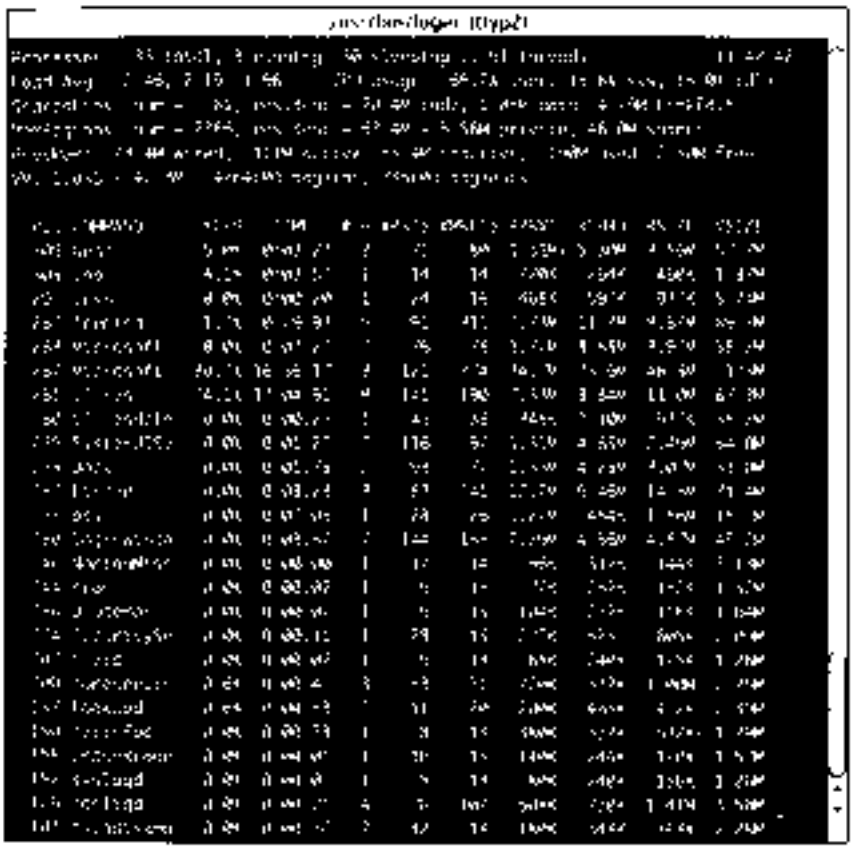


Figura 1. *El comando* **top** *nos muestra mucha información sobre nuestro sistema en pantalla. Entre otras cosas, podemos ver un listado de procesos, cuántos recursos del sistema consume cada uno, cuánta memoria libre nos queda, etc.*

Para salir de esta aplicación, sólo tenemos que presionar la tecla **Q**.

Procesos en primero y segundo plano

En el modo usuario (el modo terminal), el usuario puede tener un programa corriendo bajo dos modalidades diferentes: primer plano y segundo plano. Los procesos que se ejecutan en primer plano son aquellos que se ven en pantalla y que están interactuando con el usuario (esperando alguna respuesta del teclado, por ejemplo). Los procesos que se ejecutan en segundo plano lo hacen a trasfondo, funcionan solos, y generalmente no requieren que el usuario intervenga en su funcionamiento. El usuario puede tener sólo un proceso en primer plano (sólo un programa en pantalla, claro) y muchos procesos en segundo plano, funcionando independientemente. Cabe destacar que cualquier proceso puede estar tanto en primer plano como en segundo plano, no es necesario que el programa cumpla con ciertas características definidas. Lo único que se debe tener en cuenta es que no conviene tener un procesador de textos en segundo plano, ya que no podremos interactuar con él, y el programa no es aún lo suficientemente inteligente como para generar sus propios documentos (¿algún día llegará esa tecnología?). Algunos ejemplos de procesos que son útiles en segundo plano pueden ser un sistema de indexación de archivos, un buscador, un sistema de búsqueda de virus, etc. Todos programas que, una vez ejecutados, funcionan por su cuenta, sin la participación del usuario en tiempo real.
Cuando lanzamos un programa, éste se ejecuta directamente en primer plano. Una vez ejecutado, si queremos enviarlo al segundo plano, sólo tenemos que presionar la combinación de teclas **CTRL+Z**. Aparecerá un mensaje en pantalla, similar al siguiente:

```
[1]  +  345 Stopped                 sleep 50
```

El primer número (el que está entre corchetes) es el número de proceso que se encuentra en segundo plano. El segundo número (345) es el PID de ese proceso. Por último, el sistema nos informa el estado del proceso y cuál fue el comando que lanzó dicho proceso. Como pueden ver, el proceso está "parado" o "suspendido". Lo que debemos hacer es indicarle al sistema que el proceso comience a trabajar en segundo plano. Para esto, simplemente usamos el comando **bg** seguido del número de proceso en segundo plano que queremos que comience o continúe su trabajo.

```
[localhost:/bin] hfazena% bg 1
[1]   sleep 50 &
[localhost:/bin] hfazena%
```

El sistema ahora nos informa que el proceso ya está funcionando correctamente en segundo plano.

Otra forma de lanzar un proceso en segundo plano directamente desde el momento en que se ejecuta es agregando el símbolo '&' al final del comando de dicho proceso. Por ejemplo:

```
emacs &
```

Ejecutará el editor de textos *emacs* en segundo plano (como mencionamos anteriormente, no es muy útil ejecutar un editor de textos en segundo plano, pero vale como ejemplo). Cabe destacar que, enviando los procesos a segundo plano mediante este sistema, no hace falta iniciarlos, ya que automáticamente comenzarán a trabajar en el segundo plano.
Para traer un proceso de segundo plano al primer plano, usamos el comando **fg**, seguido del número de proceso en segundo plano. Ahora bien, ¿cómo sabemos qué procesos tenemos funcionando en segundo plano? Sencillo: usando el comando **jobs**, sin parámetros.

```
[localhost:/bin] hfarena% jobs
[1]   + Running       sleep 50
[2]   - Running       emacs
[localhost:/bin] hfarena%
```

En el ejemplo, si queremos traer *emacs* al primer plano, sólo tenemos que ejecutar el comando **fg 2**.
Los procesos en primero y segundo plano son un recurso que puede ser utilizado en casi todos los shells UNIX (bash, tcsh, csh, zsh, etc.), y resultan de gran utilidad cuando sólo se puede usar una terminal en el servidor al cual se tiene acceso.

Cambiando la prioridad de los procesos

Todos los procesos tienen un valor de prioridad de ejecución definido entre -20 y 20. Cuanto más pequeño sea el valor, más rápido se ejecutará ese proceso. En realidad, cuanto más pequeño sea el número, el procesador le dedicará más tiempo.
Es posible cambiar este valor (¿qué valor no se puede alterar en GNU/Linux?). Los usuarios normales pueden modificar la tabla de prioridades, siempre y cuando aumenten el valor de prioridad (postergación de procesos); nunca podrán hacer que un proceso se ejecute antes que otro que estaba delante de él.
Sólo el administrador puede modificar este valor en cualquiera de sus direcciones.
Para obtener un listado de procesos y prioridades, se utiliza el comando **ps -l**. Este comando presentará la siguiente información en pantalla:

```
F   UID   PID  PPID  PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
100 0     1    0     0    0     1104  368 do_sel  S    ?          0:03 init [3]
040 0     2    1     0    0     0     0   bdflus  SW   ?          0:00 [ttuld]
040 0     3    1     0    0     0     0   kupdat  SW   ?          0:00 [data]
040 0     4    1     0    0     0     0   kpiod   SW   ?          0:00 [kpiod]
040 0     5    1     0    0     0     0   kswapd  SW   ?          0:01 [kswapd]
040 0     6    1     -20  -20   0     0   md_thr  SW<  ?          0:00 [covery]
140 1     220  1     0    0     1196  288 do_sel  S    ?          0:00 portmap
040 0     273  1     1    0     1156  480 do_sel  S    ?          0:01 syslogd
140 0     284  1     0    0     1416  672 do_sys  S    ?          0:00 klogd
040 2     300  1     0    0     1128  296 nanosl  S    ?          0:00 /sbin/a
140 1     221  1     0    0     1196  288 do_sel  S    tty1       0:00 freenet
040 0     274  1     1    0     1156  480 do_sel  S    tty2       0:01 airbase
140 0     285  1     0    0     1416  672 do_sys  S    tty3       0:00 cusseme
040 2     306  1     0    0     1128  296 nanosl  S    tty4       0:00 uptime
040 0     316  1     0    0     1304  540 nanosl  S    ?          0:00 -crond
140 0     332  1     0    0     1124  436 do_sel  S    tty5       0:00 gnucad
140 0     359  1     0    0     1176  436 do_sel  S    ?          0:00 eesd
040 0     416  1     0    0     1304  540 nanosl  S    tty7       0:00 nmap
140 0     532  1     0    0     1124  436 do_sel  S    ?          0:00 inetd
140 0     459  1     0    0     1176  436 do_sel  S    ?          0:00 cpd
040 0     216  1     0    0     1304  540 nanosl  S    ?          0:00 joe
040 0     616  1     0    0     1304  540 nanosl  S    ?          0:00 crond
140 0     732  1     0    0     1124  436 do_sel  S    ?          0:00 emacs
140 0     859  1     0    0     1176  436 do_sel  S    ?          0:00 vi
140 0     432  1     0    0     1124  436 do_sel  S    ?          0:00 pico
140 0     159  1     0    0     1176  436 do_sel  S    !?         0:00 wordstar
```

Salida del comando ps ax -l.

Tengan en cuenta que la columna **NI** es la que indica el valor de prioridad.
Para cambiar el valor de prioridad se utiliza el comando **renice** en el siguiente formato:

ELIMINACIÓN DE ENLACES

Cabe destacar que la eliminación de un enlace no significa, de ninguna manera, la eliminación del archivo al cual apunta.

ACOSTÚMBRENSE AL PS

Es bueno tomar la costumbre de pedir un listado de procesos en intervalos de tiempo para ver cómo está funcionando nuestro sistema.

```
renice numerodeprioridad -p pid
```

Manejo de usuarios

Como fue mencionado, GNU/Linux es un sistema operativo **multiusuario**. Esto quiere decir que en una única instalación de GNU/Linux pueden existir múltiples perfiles de usuario. Cada usuario dispone de su propio directorio personal, en el cual puede almacenar su configuración personalizada, sus documentos, sus archivos personales, etc.
Este directorio personal se guarda dentro del directorio **/home**, y es creado automáticamente cuando el administrador da de alta al nuevo usuario. Generalmente, el nombre del directorio personal del usuario es el mismo que su nombre de usuario en el sistema, aunque esto puede ser alterado más tarde por el administrador.
En los sistemas UNIX existe una base de datos que contiene todos los usuarios del sistema, así como algunos datos personales, el directorio personal y el intérprete de comandos que utiliza. Esta base de datos se encuentra en el archivo **/etc/passwd**, y si visualizan este archivo en pantalla (**cat /etc/passwd**), obtendrán un listado similar al siguiente:

```
root:*:0:0:System Administrator:/root:/bin/tcsh
daemon:*:1:1:System Services:/var/root:/noshell
www:*:70:70:World Wide Web Server:/Library/WebServer:/noshell
unknown:*:99:99:Unknown User:/nohome:/noshell
facundo:*:500:0:Facundo Arena:/home/facundo:/bin/bash
```

Aunque parezca un poco complicada, la estructura de este archivo es bastante sencilla. Cada línea corresponde a un usuario del sistema. Veamos la última línea del listado para analizar cada uno de sus componentes. El primer campo (**facundo**) define el nombre de usuario en el sistema. Cada campo está separado por dos puntos (":"). El próximo campo ("*") quedó por una cuestión histórica, ya que originalmente allí iba la clave del usuario encriptada. Actualmente, la clave se almacena en otro archivo (**/etc/shadow**), que posee la misma estructura que éste que estamos analizando, pero cuenta con un sistema de encriptación mucho más avanzado. Por eso, en este lugar sólo vemos un símbolo asterisco. Los próximos dos campos corresponden al **UID** (User ID o número de usuario) y **GID** (Group ID o número de grupo). El UID es único e irrepetible; el GID, obviamente, no. El próximo campo define el nombre completo del usuario, y puede contener espacios. Por último, se definen el directorio personal del usuario y cuál será su intérprete de comandos por defecto.

Una idea: podemos crear usuarios para aplicaciones especificas. Por ejemplo, si queremos que el usuario "darkstar" sólo pueda usar un navegador web en modo texto, lo único que tenemos que hacer es modificar el **/etc/passwd** para que, en lugar de ejecutar **/bin/bash** como intérprete de comandos, ejecute **/usr/bin/lynx** (la ruta al ejecutable de *Lynx*, un navegador web para el modo texto). La próxima vez que el usuario "darkstar" entre en el sistema, Lynx será ejecutado automáticamente y será el único programa que podrá utilizar. Una vez que haya finalizado su utilización, el sistema volverá automáticamente a la pantalla de inicio de sesión.

Creación de un nuevo usuario

Para crear un nuevo usuario en el sistema, tenemos dos alternativas posibles. La primera de ellas es utilizar el método manual, el cual consiste en agregar una nueva línea al archivo **/etc/passwd** (respetando el orden de los campos), crear un directorio para el usuario dentro de **/home**, y luego asignar una clave a ese usuario con el comando **passwd [usuario]**.
La otra forma es usar directamente el comando **adduser** seguido del nombre de usuario que se va a crear. Esto hará todo el trabajo de forma totalmente automática. En algunos sistemas, **adduser** también asigna una clave de acceso, pero si no se la pide, tendrán que asignarla ustedes mismos haciendo uso del comando **passwd**.

Eliminación de un usuario

El proceso de eliminación de un usuario no es muy común en los sistemas GNU/Linux, ya que nunca se sabe si el usuario puede llegar a volver. Pero si aun así lo que desean es eliminar ese perfil de usuario de su sistema, todo lo que deben hacer es borrar su entrada correspondiente en el archivo **/etc/passwd**, **/etc/shadow** (posee la misma estructura que el anterior) y eliminar su directorio personal de **/home**.

Deshabilitación temporaria de usuarios

El administrador del sistema puede llegar a necesitar la deshabilitación temporaria de un usuario, sea por vacaciones o por cualquier otro motivo que realmente no nos interesa. Hay varias alternativas posibles; una es cambiarle la clave (usando el comando **passwd**). Pero el individuo en cuestión puede llegar a sentirse ofendido cuando note que no puede ingresar en el sistema con su clave. Otra opción es cambiar su intérprete de comandos (definido en **/etc/passwd**) por un script que muestre un pequeño mensaje en pantalla y luego vuelva a la pantalla de inicio de sesión. El script puede ser similar al siguiente:

```
#!/usr/bin/tail +2
Usted ha sido temporariamente deshabilitado de este sistema.
```

Este script puede estar ubicado en **/usr/sbin** (el directorio de las aplicaciones del administrador), y recuerden que debe tener permisos de ejecución (**chmod +x [archivo]**).

Algunos comandos relacionados con los usuarios

Es posible saber qué usuarios están conectados a nuestro sistema en un momento dado. Para esto disponemos de dos comandos: **who** y **users**. Los dos son bastante similares, con la diferencia de que **who** muestra más información que **users**. Ambos comandos pueden ser usados sin parámetros.

```
localhost% users
hfarena trax darkstar
localhost%
```

Lo único que nos indica **users** es el nombre de los usuarios que están conectados. Si queremos algo más de información (como saber en qué terminal está cada usuario), entonces tendremos que usar el comando **who**.

```
localhost% who
hfarena      tty1    Jun 5 10:08
trax         tty2    Jun 5 10:10
darkstar     tty3    Jun 4   4:40
localhost%
```

MODIFICACION DE /ETC/PASSWD

Como casi todos los archivos localizados en /etc, passwd es un archivo de texto que puede ser modificado con cualquier editor de textos.

ELIMINACION DE PROCESOS

Existe una forma de eliminar procesos que se encuentran en primer plano (los procesos de usuario que están en pantalla interactuando con el usuario). Ésta consiste en presionar la combinación de teclas CTRL+C.

Ejecución de aplicaciones

Ejecutar aplicaciones en GNU/Linux es un proceso muy sencillo. Sólo hay que tipear su comando correspondiente. Ahora bien, si el binario ejecutable de esta aplicación está en un directorio incluido en la variable de entorno **$PATH**, no hay ningún problema. Pero si está en otro directorio, entonces tendremos que dirigirnos a ese directorio y ejecutar el binario precedido por un simbolo "./". Cabe destacar que, generalmente, en la variable **PATH** se encuentran solamente los directorios del sistema. Las aplicaciones suelen almacenar su binario ejecutable en **/usr/bin**, el cual es un directorio que está incluido en el **PATH**. Pero si por algún caso la aplicación necesita ser almacenada en otro directorio, entonces habrá que usar la sintaxis **./[programa]** para ejecutarlo, desde el directorio de esa aplicación.

Editores de texto

Numerosas veces se encontrarán con la necesidad de editar algún archivo de configuración o algún script. Por esta razón, es de vital importancia que adopten el uso de una de estas herramientas. En el mundo de GNU/Linux (y los demás UNIX en general), el terreno está dominado por dos grandes: emacs y VI. Lean la siguiente sección, y luego hagan su elección.

El editor GNU: emacs

Emacs es un editor de textos originalmente creado por Richard Stallman (padre del proyecto GNU y de la Fundación del Software Libre). Sus siglas significan "Editor MACros", porque, originalmente, su fuente era la de poder desarrollar macros de programación para automatizar tareas. Actualmente, decir que *emacs* es sólo un editor de textos es realmente una falsedad. En realidad, *emacs* es todo un entorno de trabajo que incluye editor de textos, sistema de desarrollo con soporte para los lenguajes más importantes, navegador de Internet, sistema de programación en Lisp, cliente de e-mail, noticias, juegos, documentación y mucho, pero mucho más.
Para ejecutar *emacs* desde una consola en modo texto, sólo hay que tipear el comando **emacs**, y éste aparecerá en pantalla.
Pero no hay menús ni funciones con el mouse. Por esta razón, si son principiantes, lo mejor es ejecutarlo desde XWindow. Se busca en el menú **Utilidades** del administrador de ventanas favorito bajo el nombre de **XEmacs** o, si no, simplemente se abre una terminal virtual y se tipea el comando **emacs**.

XEmacs es la versión de *emacs* para XWindow. Luce tan sencillo como la versión original, sólo que nos da la posibilidad de usar el mouse y navegar por todas sus funciones mediante un sencillo sistema de menús descolgables. Permite trabajar con varios archivos al mismo tiempo (*Buffers*), y podemos pasar de uno a otro con sólo cliquear en la entrada correspondiente en el menú **Buffers** (la primera opción del menú superior). También tiene funciones muy interesantes, como **Files/Split Window**, que divide la pantalla en dos ventanas y nos permite tener un archivo abierto en cada una; y **Files/Make new frame**, que crea una ventana nueva con el archivo que estamos editando actualmente, donde, si queremos, podemos abrir un nuevo archivo. En el menú **Tools** van a encontrar muchas herramientas. Si seleccionan **Tools/Display Calendar**, verán que en la parte de abajo de la pantalla aparecerá un bonito calendario que no perturbará la visión de la ventana de edición en absoluto. Una vez en el calendario, cliqueen sobre su ventana y verán cómo las opciones principales del menú son más, y permitirá ver las fechas festivas, las fases de la Luna, y hasta mantener un diario personal.

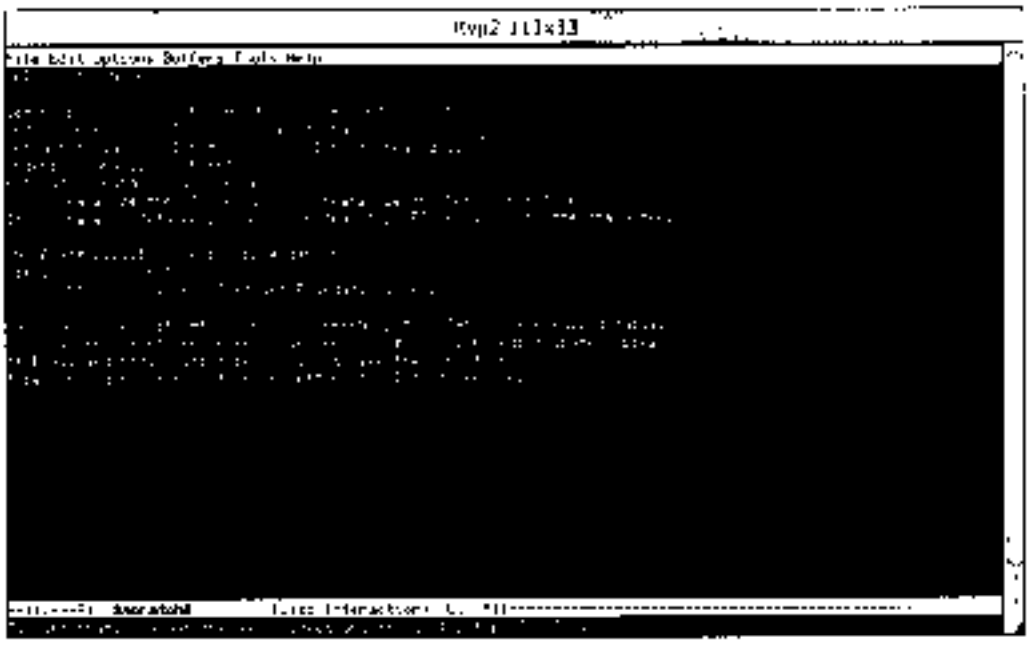


Figura 2. emacs es uno de los editores de textos más potentes para el entorno UNIX.

Vim: la evolución de un clásico

La versión de GNU/Linux de este editor se llama Vim, con una "m" al final por "Vi Improved" o "Vi mejorado". Se trata de un potente editor de textos que, como mencionamos anteriormente, tiene muchos fanáticos en el mundo de UNIX y GNU/Linux. Para ejecutarlo, sólo hay que tipear **vi** desde la línea de comandos. Verán que aparecerá una pantalla negra de bienvenida, y luego, con sólo apretar la tecla "i" entrarán en modo edición y podrán comenzar a escribir en el archivo. Ahora bien, casi

todas las opciones se manejan mediante la secuencia de teclas **ESC** + ":". Por ejemplo, para guardar un archivo hay que tipear **ESC** + ":" + "**w**", o para salir a la consola, **ESC** + ":" +."**q**". Esto resulta más que tedioso para la mayoría de los nuevos usuarios, por lo que una interfaz gráfica es lo más recomendable. Para eso está **gVim**.
gVim es un entorno gráfico para Vim desarrollado bajo las librerías GTK (las usadas para GNOME) que nos permitirá acceder a todas las funciones de Vim mediante un sencillo sistema de menús descolgables, igual que XEmacs.
La interfaz es mucho más amigable que la de XEmacs: colores, iconos y ventanas de diálogo hacen que gVim parezca una herramienta nativa de XWindow. Eso sí, no van a encontrar la cantidad de "agregados" que hay en XEmacs, como el calendario o el visor de correo electrónico. gVim es un editor de textos con todas las letras, y su funcionalidad apunta directamente a cumplir con esa función.

Figura 3. *Así como hay muchos seguidores de emacs, hay muchos seguidores de Vi. Bonita interfaz, ¿no?*

Los niveles de ejecución

Luego de la carga y la posterior ejecución del *Kernel* de Linux, el sistema comienza la ejecución de una serie de procesos elementales para su correcto funcionamiento. El responsable de esta etapa de inicialización es el proceso **init**.
init basa su ejecución en un archivo localizado en **/etc/inittab**, en el cual se encuentran detallados los diferentes niveles de ejecución (*run levels*) y sus procesos.
Éste es un archivo **/etc/inittab** de ejemplo:

```
Archivo de configuracion /etc/inittab
#
# inittab       Este archivo describe cómo el proceso init debe configurar
#               el sistema en un nivel de ejecución determinado.
#
# Autor:        Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modificado para RHS Linux por Marc Ewing y Donnie Barnes
#
# runlevels predeterminados. Los runlevels que utiliza RHS son:
#   0 - Parado (no configure éste como predeterminado)
#   1 - Modo Monousuario
#   2 - Multiusuario, sin NFS (Lo mismo que el 3 si no tiene una red)
#   3 - Modo multiusuario completo
#   4 - sin usar
#   5 - X11
#   6 - reiniciar (no configure éste como predeterminado)
#
id:3:initdefault:

# Inicialización del sistema
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Cosas que se ejecutan en cada nivel de ejecución.
ud::once:/sbin/update

# Atrapar el CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Cuando nuestro UPS nos dice que la alimentación ha fallado, asumimos que
# tenemos unos minutos de alimentación más. Registramos un cierre de sistema
# de 2 minutos por ahora.
# Esto, obviamente, asume que usted tiene un UPS conectado y trabajando
# correctamente.
```

```
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# Si la alimentación se restableció antes de que se ejecute
shutdown,cancelamos.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"


# Ejecutar gettys en los niveles de ejecución estándar.
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Ejecutar xdm en el runlevel 5
# xdm ahora un servicio separado.
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Este archivo es, básicamente, un *script*. Las líneas que comienzan con un # (símbolo numeral) son comentarios y no tienen efecto sobre la ejecución del sistema.
La primera línea sin comentar le indica a **init** que el nivel de ejecución configurado por defecto es el 3. Al igual que la primera, las demás líneas tienen todas el mismo formato: valores separados por signos ":".
El primero es el valor identificador del proceso (utilizado internamente por **init**), el cual deberá ser un valor numérico. El segundo es la secuencia de números correspondientes a los diferentes niveles en los que se quiere ejecutar ese proceso. El tercer parámetro es la acción que se realizará con ese proceso. Por ejemplo, si en ese campo se encuentra la cadena *wait*, **init** ejecuta el proceso configurado y espera a que termine. El último valor de la línea es el proceso en cuestión que se ejecutará.
Las acciones que pueden realizarse sobre los procesos son las siguientes:

OPCIÓN	DESCRIPCIÓN
respawn	Si el proceso no existe, lo crea y no espera a que termine (continúa con la ejecución del **inittab**). Si el proceso existe, no hace nada. Cuando el proceso muere, se vuelve a ejecutar.
wait	Ejecuta un proceso y espera a que termine.
once	Crea el proceso y continúa con la lectura del **inittab**. Cuando muere, no vuelve a ser ejecutado.

OPCIÓN	DESCRIPCIÓN
boot	Crea el proceso solamente durante la etapa de booteo. No espera a que termine, y cuando muere no vuelve a ser ejecutado.
bootwait	Se crea el proceso cuando se realiza el cambio entre modo monousuario y modo multiusuario. Si el nivel de ejecución configurado por defecto es el 2, entonces se creará justo después del booteo.
powerfail	Ejecuta el proceso solamente cuando init recibe una señal de falla de alimentación (SIGPWR).
powerwait	Ejecuta el proceso solamente cuando init recibe una señal de falla de alimentación, y espera a que concluya antes de continuar con los demás procesos.
off	Si el proceso se está ejecutando, envía una señal de advertencia (SIGTERM) y espera 20 segundos antes de enviarle la señal de muerte (SIGKILL). En caso de que el proceso no exista, esta acción es ignorada.
initdefault	Define el nivel de ejecución en que se iniciará el sistema por defecto.
sysinit	Los procesos que lleven esta acción se ejecutarán antes de que init intente acceder a la consola para iniciar la sesión de login.

Veamos una sección importante del archivo:

```
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
```

Esta sección ejecuta todos los archivos **rc** del directorio **/etc/rc.d**.
Estos archivos son scripts que ejecutan diferentes tareas de inicialización del sistema. Más adelante trabajaremos con el archivo **/etc/rc.d/rc.local** para la ejecución de nuestros propios programas en el arranque.
La última línea del archivo ejecuta el *X Display Manager* en el nivel de ejecución 5. Este programa se encarga de presentar una pantalla gráfica de inicio de sesión o *login*. La configuración de este programa se encuentra en **/etc/X11/xdm**, pero si ustedes tienen instalada alguna de las distribuciones más famosas, no será necesario hacer modificación alguna. Luego de la registración, XDM ejecuta XWindow con el administrador de ventanas configurado por defecto.

El comando init

Para cambiar a otro nivel de ejecución, no necesariamente hay que reiniciar el sistema. En Linux, existe el comando **init**, el cual puede ser ejecutado desde la linea de comandos y tiene el siguiente formato:

```
init runlevel
```

El parámetro *runlevel* es un valor numérico que especifica el nivel de ejecución (entre 0 y 6). Por ejemplo, si se ejecuta **init 5**, el sistema cambiará el nivel de ejecución a 5 (modo X11), que ejecutará automáticamente **xdm**.
Para obtener más información sobre la sintaxis del archivo de configuración **/etc/inittab**, tipeen desde la línea de comandos de su *shell*: **man inittab**.

Instalación de aplicaciones

En el famoso sistema operativo masivo, predomina sólo un formato de compresión de paquetes: el ZIP. El hecho de que haya un único formato es, en cierta forma, una ventaja, ya que los usuarios pueden acostumbrarse a descomprimir todos los programas usando una misma herramienta. Pero esto, a la vez, es una desventaja, ya que nadie asegura que ese formato sea el mejor para todas las ocasiones, y, consecuentemente, los usuarios no tienen opciones para elegir.
En GNU/Linux, existen varios formatos de compresión y empaquetado. Los que dominan el universo son básicamente tres: **TGZ**, **RPM** y **DEB**. Si bien los dos últimos son bastante similares entre sí, las técnicas por seguir para instalar cada formato son bastante diferentes. Vamos a repasar la filosofía de cada formato y las formas de instalar/desinstalar cada uno.

El formato TGZ

Este formato tiene ya varias décadas de vida en el mundo de UNIX. Actualmente, la única distribución que lo utiliza para empaquetar sus aplicaciones es Slackware. Pero también se utiliza mucho para distribuir las aplicaciones en código fuente. TGZ es un formato generado por dos programas: el primero, TAR, se encarga de concatenar el contenido de varios archivos para armar un solo paquete. El segundo programa, GZIP, tiene la función de comprimir ese paquete generado por TAR. La extensión de estos paquetes es **.tar.gz** o **.TGZ**. Ambos poseen una estructura interna idéntica.

Para descomprimir un paquete en este formato, realicen los siguientes pasos:

1. Copiar el paquete a un directorio temporal.
2. Descomprimirlo y desempaquetarlo con el siguiente comando:

```
tar -xzvf archivo.tar.gz
```

o

```
tar -xzvf archivo.tgz
```

Automáticamente se creará el árbol de archivos y directorios correspondiente a ese paquete.
Ahora bien, sabemos que este formato se usa mucho para distribuir el código fuente de aplicaciones. Para compilar una aplicación hay que (generalmente) seguir estos pasos desde el directorio del código fuente:

1. Configurar el código fuente con el comando **./configure**.
2. Compilar (comando **make**).
3. Instalar los archivos compilados (**make install**).

Se preguntarán cómo ejecutar el programa. Este tipo de paquetes generalmente contiene un directorio llamado **/src**, en donde se ubica el código fuente principal, y en donde se almacena el archivo compilado finalmente. Busquen ese archivo una vez que hayan finalizado el proceso de compilación, y ya sabrán cómo ejecutar el programa. Para reconocerlo, noten que tiene las propiedades de un archivo binario ejecutable (aparece en verde y posee un asterisco a su lado cuando hacen un listado de directorios).
Para desinstalar un paquete **.tar.gz**, en lugar de ejecutar **make install**, deben ejecutar **make uninstall**. Obviamente, podrán hacer esto siempre y cuando dejen el directorio del código fuente de la aplicación intacto. Si lo borran (lo más común), entonces el proceso será más complicado, ya que habrá que ir borrando archivo por archivo.
Si usan la distribución Slackware, disponen de una herramienta que les simplificará el proceso de instalar y desinstalar una aplicación: para instalar un paquete TGZ (o **.tar.gz**) deben tipear el comando:

```
installpkg archivo.tar.gz
```

o

```
installpkg archivo.tgz
```

El sistema se encargará de todo automáticamente. Para desinstalar un paquete:

```
removepkg archivo.tar.gz
```

También disponen de una herramienta llamada **pkgtool**, que les permitirá instalar y remover aplicaciones mediante un sencillo sistema de menús.

El formato RPM

Este formato fue desarrollado originalmente por Red Hat (de ahí las siglas *Red Hat Package Manager*), y es muy superior a la combinación TAR-Gzip. Esto es básicamente porque el formato RPM posee una base de datos central que registra todos los programas instalados, con sus respectivas versiones. De esta forma, el proceso de instalación o actualización de un programa es mucho más sencillo. Actualmente, RPM es el formato elegido por todas las distribuciones "grandes", como Red Hat, SuSE, Conectiva y Mandrake.
Antes de instalar un RPM, comprueben en el sitio o en la documentación que se trate de un paquete para la distribución que están usando. Este formato se usa para distribuir los programas en forma binaria (ya compilada). Por esta razón es importante que instalen los paquetes RPM que contienen archivos compilados para la distribución que usan.
Para instalar un paquete RPM, copien el archivo a un directorio temporal y tecleen:

```
rpm -i archivo.rpm
```

donde archivo es, generalmente, el nombre del programa. El siguiente es un ejemplo:

```
rpm -i licq-2.2.2.rpm
```

Para desinstalar un programa instalado previamente desde un paquete RPM, sólo hay que tipear:

```
rpm -e (nombre_del_programa)
```

El nombre del programa no incluye la versión. Para nuestro ejemplo del **licq**, la línea quedaría como la siguiente:

```
rpm -e licq
```

El sistema RPM se encargará automáticamente de remover todos los archivos de la aplicación y de quitar su correspondiente registro de la base de datos RPM.
Muchas veces sucede que el sistema RPM no encuentra ciertas librerías o componentes que, en realidad, sí están instalados, y no permite descomprimir el paquete. Para esto, existe un último recurso llamado "fuerza bruta" (úsenlo bajo su propio riesgo):

```
rpm --force --nodeps archivo.rpm
```

Esos parámetros harán que RPM no revise las dependencias de la aplicación y sobrescriba todos los archivos que ya se encuentran instalados en el sistema con los que están contenidos en el paquete.
Muchos se preguntan cómo hacer para ejecutar un programa luego de haber instalado su correspondiente paquete RPM sin ningún tipo de problemas. Hay una solución sencilla, que consiste en buscar el ejecutable dentro del paquete RPM para saber con qué comando iniciar dicho programa. *Midnight Commander* (el cual se inicia con el comando **mc**) nos permite navegar por dentro del contenido de un archivo RPM (además de los TGZ, los ZIP y los DEB). Sólo hay que presionar **ENTER** sobre el paquete RPM, seleccionar la opción **CONTENTS.cpio** y luego, navegar por el listado de directorios de dicho paquete. Recuerden que los binarios ejecutables generalmente se almacenan en **/bin**, **/usr/bin** o **/usr/share/bin**.

El formato DEB

Los usuarios de Debian gozan de la aplicación **dselect** para instalar y desinstalar aplicaciones. Este sistema controlado por menús es muy sencillo de usar, así que no perderemos tiempo explicando su utilización.
También disponen de otra herramienta, muy famosa, llamada **dpkg**. Su utilización es muy sencilla y similar a la de RPM. Para instalar un paquete sólo tienen que tipear:

```
dpkg -i archivo.deb
```

Eso será más que suficiente para descomprimir el paquete y copiar su contenido a los lugares correspondientes. Cabe destacar, también, que pueden instalar y navegar por el contenido de los paquetes desde Midnight Commander (**mc**), de la misma forma que con los paquetes RPM.

Para desinstalar un paquete Deb, tienen que usar la utilidad **dpkg** del siguiente modo:

```
dpkg -r nombre_del_paquete
```

Recuerden que el nombre del paquete no es el nombre del archivo. Generalmente es el nombre del programa seguido de su versión.

Otros formatos

Si quieren descomprimir archivos empaquetados y comprimidos bajo el algoritmo de compresión ZIP, sólo tienen que usar la herramienta **unzip** del siguiente modo:

```
unzip archivo.zip
```

El arranque de GNU/Linux

GNU/Linux concentra todo su poderío en un núcleo monolítico comúnmente denominado *Kernel*. Éste, al aumentar cada día más su tamaño, debe ser cargado por algún programa más pequeño que quepa en un sector booteable estándar y que se encargue de gestionar el proceso de arranque. El gestor oficial de Linux es LILO.
LILO nos permite bootear diferentes sistemas operativos, incluso si éstos están instalados en la misma máquina.
Se carga en la *MBR* de la unidad de disco principal, y presenta un *prompt* que nos permite seleccionar el sistema operativo que se iniciará, por medio de etiquetas.

El archivo /etc/lilo.conf

Todos los datos de configuración de LILO se almacenan en este archivo, el cual no es más que un archivo con secciones y sus respectivos valores. Si buscan este archivo en su sistema, debe asemejarse al siguiente:

```
10: boot = /dev/hda
20: prompt
30: timeout = 50
```

```
40: default = Linux
50: map=/boot/map
60: install=/boot/boot.b

image = /boot/bzImage
label = Linux
root = /dev/hda2

other = /dev/hda1
label = dos
table=/dev/hda
```


Este archivo de ejemplo tiene definidos dos sistemas operativos: Linux y DOS.
La primera sección contiene la configuración general de LILO: en la línea 10 definimos la unidad principal en la que se instalará LILO. En la línea 20 indicamos que presente un prompt y espere (timeout 50). En la línea 30 definimos el tiempo que LILO mostrará el prompt y esperará por una respuesta. La línea 40 contiene el sistema operativo que se cargará por defecto.
Las líneas 50 y 60 llevan las rutas de los archivos **System.map** y **boot.b** correspondientes al *kernel*. Estos archivos generalmente se encuentran en el directorio **/boot**. La segunda sección contiene el detalle de los sistemas operativos y las particiones que se bootearán.

Cómo agregar un nuevo kernel Linux

Cuando se "juega" demasiado con el *kernel* principal, es conveniente tener otro a mano por si éste no puede ser cargado.
Para configurar el nuevo kernel de GNU/Linux, comúnmente se procede a agregar las siguientes líneas al archivo **lilo.conf**:

```
image = /boot/bzImage
label = Linux
root = /dev/hda2
```

En la primera línea se define la ruta completa de la imagen del kernel de Linux. Cuando se termina el proceso de compilación del kernel, se genera un archivo imagen, el cual deber ser copiado al directorio **/boot**. Para obtener más información acerca de la compilación del kernel, lean el **Apéndice**.
En la segunda línea, definimos la etiqueta con la cual invocaremos el sistema operativo. Por último, en la tercera línea, definimos la partición principal del sistema.

Agregar otros sistemas operativos

Para configurar otros sistemas operativos, el proceso es similar. Se agregan las siguientes líneas al archivo de configuración de LILO:

```
other = /dev/hda1
label = dos
table=/dev/hda
```

En la primera línea definimos como valor la partición en la que se encuentra ese otro sistema operativo. En las últimas dos líneas, definimos la etiqueta y la unidad en que reside la tabla de particiones.

El archivo /etc/rc.d/rc.local

Muchas veces surge la necesidad de ejecutar periódicamente un comando, cada vez que se enciende el sistema. Para esto, tenemos que definir en el archivo **/etc/rc.d/rc.local** qué comando es el que queremos ejecutar.
Algunas distribuciones ya incluyen muchos comandos que realizan diferentes tareas; lo mejor será incluir nuestros propios comandos al final del archivo con nuestro editor de textos preferido.
En los capítulos avanzados del libro haremos un uso intensivo de este archivo para mantener configuraciones y realizar otras tareas.

Guía de comandos

A continuación, presentamos una guía con los comandos más importantes que se usan en el modo texto de GNU/Linux.

Comandos para el manejo de archivos

COMANDO	DESCRIPCIÓN
cd	Cambia de directorio.
compress	Comprimir archivos.
cp	Copiar un archivo.
chmod	Cambia los permisos de un archivo o directorio.

COMANDO	DESCRIPCIÓN
chown	Cambia el propietario del archivo o directorio.
df	Muestra el espacio libre en el disco.
du	Muestra el espacio en el disco utilizado.
fdformat	Formatea un disquete.
fdisk	Utilidad para el particionamiento de unidades.
find	Encontrar un archivo.
fsck	Revisión del sistema de archivos.
gzip	Descomprime un archivo en formato gzip
ln	Crea un enlace simbólico.
ls	Muestra el contenido de un directorio (análogo a dir).
mkdir	Creación de un directorio.
mkfs	Crea un nuevo sistema de archivos.
mknod	Crea un archivo de dispositivo.
mkswap	Crea un espacio de intercambio.
mount	Montar una unidad/partición en el sistema de archivos.
mv	Mover un archivo (se utiliza también para renombrar).
pwd	Devuelve la cadena correspondiente al directorio actual.
rm	Borrar un archivo.
rmdir	Borrar un directorio.
split	Dividir un archivo en muchos archivos de un tamaño determinado
swapoff	Desactivar el espacio de intercambio.
swapon	Activar el espacio de intercambio.
sync	Sincronizar los procesos de escritura/lectura.
tac	Da vuelta el contenido de un archivo.
tail	Muestra las últimas líneas de un archivo.
tar	Desempaquetar un archivo en formato tar.
umount	Desmontar un sistema de archivos ya montado.
uniq	Borra las líneas duplicadas de un archivo.
wc	Cuenta el número de caracteres de un archivo.

Comandos para el manejo de procesos

COMANDO	DESCRIPCIÓN
bg	Ejecutar un proceso (interrumpido) en segundo plano
fg	Ejecutar un proceso (interrumpido) en primer plano.
free	Muestra la memoria libre y utilizada.
halt	Apagar la máquina.
kill	Enviar una señal a un proceso.
ldd	Muestra las librerías que se necesitan para ejecutar un programa.
nice	Ejecutar un proceso con menor prioridad de ejecución.

COMANDO	DESCRIPCIÓN
ps	Muestra todos los procesos que se están ejecutando en el sistema.
pstree	Como el ps, pero muestra todo en forma de árbol.
reboot	Reiniciar el sistema.
shutdown	Cerrar el sistema.
top	Una utilidad para monitorear procesos y el estado del sistema.
uname	Muestra información del sistema.

Comandos para el manejo de usuarios

COMANDO	DESCRIPCIÓN
adduser	Crear un nuevo usuario.
chsh	Cambiar el shell de un usuario determinado.
groups	Muestra el listado de grupos de usuarios del sistema.
id	Muestra la información de usuario y del grupo de un determinado usuario.
passwd	Cambiar la clave de acceso de un determinado usuario
su	Cambiar de usuario.
users	Lista los usuarios conectados al sistema.
who	Muestra información de los usuarios actualmente conectados.

Otros comandos

COMANDO	DESCRIPCIÓN
alias	Permite ejecutar comandos utilizando otros nombres (crea alias).
apropos	Encontrar las páginas de manual para un comando determinado.
cal	Muestra el calendario.
date	Permite ver y cambiar el día y la hora actuales.
info	Muestra ayuda de un determinado comando.
man	Muestra las páginas del manual de un determinado comando.
unalias	Borrar el alias de un comando.

En resumen...

En este capítulo hemos aprendido a utilizar el sistema GNU/Linux desde el modo texto, utilizando diferentes comandos para el manejo de archivos, procesos y usuarios (tres recursos fundamentales de todo sistema UNIX). Más adelante volveremos al entorno textual para hacer un uso avanzado del sistema.

LA BIBLIA DE

Capítulo 4

La programación en el lenguaje Bash

El intérprete de comandos Bash incluye un lenguaje de programación de scripts muy poderoso. Analizaremos su uso en este capítulo.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

El lenguaje Bash

Cuando un administrador precisa realizar algún pequeño programa para automatizar ciertas tareas administrativas, piensa en el lenguaje Bash. Este completo *shell* posee un lenguaje de programación muy poderoso.
Vamos a comenzar haciendo nuestro primer script en Bash.
Los scripts no son más que archivos de texto interpretados. Por esta razón, vamos a iniciar un editor de textos (**emacs**, por ejemplo), y tipearemos las siguientes líneas:

```
#!/bin/sh
# Ejemplo 1: Hola Mundo!
echo "Hola Mundo!"
```

Luego, le otorgaremos permisos de ejecución (755). Una vez concluido el proceso de desarrollo de esta pequeña aplicación, tipeen el nombre del programa:

```
./hello
```

Éste devolverá el siguiente mensaje en pantalla:

```
Hola Mundo!
```

Este pequeño programa consta de tres líneas. La primera le indica al intérprete de comandos qué intérprete debe usar para ejecutar el lenguaje que fue utilizado en el programa. La segunda es sólo un comentario (en Bash, los comentarios comienzan con el símbolo "#").
Por último, la tercera línea es el comando **echo**, con un valor de cadena encerrado entre comillas. Este comando imprime la cadena en la terminal activa.
Como pueden ver, el funcionamiento de este sistema es básicamente sencillo. Ahora vamos a adentrarnos un poco más en su desarrollo.

LENGUAJE SUPERPODEROSO

Bash posee un lenguaje de programación ideal para desarrollar scripts de automatización de tareas administrativas.

DONDE ENCONTRAR INFORMACIÓN

El comando 'man bash' les ofrecerá un completo manual de utilización del lenguaje de programación de Bash.

Variables

La definición de variables en el lenguaje Bash es igual que la definición de variables de entorno. Si queremos definir la variable "edad" con el valor "30", sólo debemos ingresar:

```
edad=30
```

Noten que no se utilizan espacios. Respeten esta propiedad, o les traerá muchos dolores de cabeza.
Ahora, si queremos imprimir la variable en pantalla, utilizamos la misma función **echo**, pero le pasamos como parámetro el nombre de la variable precedido por un símbolo "**$**".

```
echo $edad
```

También es posible incluir variables en medio de una cadena; se las llama de la misma manera. Por ejemplo:

```
echo "La edad de este usuario es: $edad años"
```

De esta manera, se imprimirán todos los caracteres encerrados entre comillas y, cuando llegue a "$edad", se imprimirá el valor de esta variable.

Paso de argumentos en la línea de comandos

Es posible pasar argumentos en la línea de comandos al *script* en ejecución. Esto significa, básicamente, que el programa puede funcionar en relación con ciertos parámetros definidos por el usuario. Para esto, Bash provee una serie de variables predefinidas.
La variable "$0" almacena el nombre del script. Las variables siguientes ("$1", "$2", etc.) guardan los diferentes parámetros. El siguiente ejemplo les permitirá aclarar estos conceptos:

```
#!/bin/sh
# archivo argumentos
# Ejemplo 2: Paso de argumentos en la línea de comandos
# Modo de uso ./argumentos [nombre] [apellido]

echo "Bienvenido a $0"
echo "Su nombre es $1"
echo "Su apellido es $2"
echo "Todas las variables sumadas: $*"
```

De este modo, si ejecutan:

```
./argumentos Facundo Arena
```

el programa devolverá:

```
Bienvenido a ./argumentos
Su nombre es Facundo
Su apellido es Arena
Todas las variables sumadas: Facundo Arena
```

Recuerden que la variable $* guarda todos los argumentos concatenados.
Este método les será útil para la creación de programas con múltiples opciones, configuradas desde la línea de comandos en el momento de ejecución. Otras variables predefinidas son:

$# Cantidad de parámetros pasados a la función.
$? Valor devuelto del último programa ejecutado dentro del script.

Ingreso de datos

Se estarán preguntando cómo hacer para que el usuario de nuestro programa pueda ingresar datos por medio del teclado. Ya llegó el momento de explicarlo: Bash utiliza el comando **read** para el ingreso de datos en variables. Su formato es:

```
read [variable]
```

Cuando la interpretación del script llegue a este punto, esperará un ingreso de datos proveniente del teclado y lo almacenará en la variable indicada. Ahora desarrollaremos el mismo ejemplo de la sección anterior, pero utilizando esta función.

```
#!/bin/sh
# Archivo pread
# Ejemplo 3: Utilización de la función read
# Uso ./pread

echo -n "Ingrese su nombre:"
read nombre
echo -n "Ingrese su apellido:"
read apellido
echo "Su nombre es $nombre"
echo "Su apellido es $apellido"
```

Como pueden apreciar, el uso de este comando es muy sencillo. Noten el parámetro -n en la función *echo*. Esto hace que luego de la impresión del mensaje, *echo* no baje de linea, sino que se mantenga allí.

Evaluación de expresiones

Bash también provee un sistema de evaluación de expresiones aritméticas. ¡Una calculadora en una linea de programación!

El formato para realizar estas evaluaciones es:

```
$[expresion]
```

Por consiguiente, la siguiente expresión es válida:

```
valor=$[25+25]
```

El resultado de *valor* será, obviamente, 50. Los operadores que habrá que utilizar en las expresiones son:

```
- +   resta y suma
!       negación
* / %  Multiplicación, división y resto.

<< >>  movimiento de bit
<= >= < > comparaciones

== !=  igualdad, desigualdad
&       AND
^       OR exclusivo
|       OR
&&      AND lógico
||      OR lógico
```

Se pueden utilizar variables dentro de las expresiones:

```
#!/bin/sh
# archivo suma
# Ejemplo 3: evaluación de expresiones aritméticas
# uso ./suma
numero1=5
numero2=20
numero3=$[numero1+numero2*100]
echo $numero3
```

También hay operadores para evaluar archivos o directorios. Éstos son:

-d Si el archivo es un directorio.
-f Si es un archivo común.
-r Si el permiso de ejecución es activado.
-s Si el largo del archivo es mayor que 0.
-w Si el permiso de escritura está activado.
-x Si el permiso de ejecución está activado.

Estructuras condicionales

Las estructuras condicionales son las que hacen interesantes a los scripts. Utilizándo las es posible crear menús con opciones, tomar diferentes decisiones sobre la base de una situación especifica, hacer preguntas, y un sinfín de acciones más. La más famosa y utilizada de estas estructuras es el if condicional. Su forma básica es:

```
if [ expresion ]
then
accion1
elif [expresion]
then
accion 2
else
accion3
fi
```

La primera linea evalúa la expresión entre corchetes. Si es verdadera, ejecuta "**accion1**"; si es falsa, salta la ejecución hasta el **else** y ejecuta "**accion2**".
Por último, se cierra la estructura con un **fi**.
Veamos ahora un ejemplo sencillo de la utilización del condicional **if**:

```
#!/bin/sh
# Ejemplo 3: Estructuras condicionales

echo -n "Ingrese un valor numérico:"
read valor
if [ $valor = 20 ]  #recuerde separar la expresión de los corchetes
then
echo "el valor es igual a 20"
else
echo "El valor ingresado es diferente a 20"
fi
```

Otra estructura condicional es **case**. Su forma básica es la siguiente:

```
case [variable] in
 opcion1 | opcion2)
  funcion1 (...) ;;
 opcion1 | opcion3)
  funcion2 (...) ;;
 opcion1 | opcion2)
  funcion2 (...) ;;
 *)
  funcion3 (...) ;;
esac
```

La estructura **case** evalúa la variable **[variable]** con las distintas opciones definidas por el usuario. Si se cumple alguna, se ejecutan las funciones correspondientes a esta opción. El simbolo "|" funciona como un "**O**", con lo que ambas opciones pueden ser válidas (no es necesario utilizar la segunda opción). Cada opción finaliza con un doble simbolo ; (;;).
Si no se cumple ninguna de las opciones, se ejecutará (siempre y cuando esté definida) la opción "*". Por último, para cerrar la estructura se utiliza **esac**.

Bucles

También podemos usar los bucles repetitivos *while* (condicional) y *for* para repetir ciertas porciones de código fuente una determinada cantidad de veces, o hasta que se cumpla una evaluación de expresión. El funcionamiento es el mismo que en los otros lenguajes de script/programación. Comencemos por la estructura del ciclo **while**:

```
while [condicion]
do
accion1
[...]
done
```

Si la condición se cumple, **while** ejecuta "accion1" y todas las operaciones siguientes hasta que la condición sea falsa.
Un ejemplo sencillo de **while** es el siguiente:

```
#!/bin/sh
#Ejemplo 4: Uso de while
opcion=100 # Ponemos opcion en 100 para que el ciclo se cumpla la primera vez.
while [ $opcion != 0 ] #recuerde separar la expresión de los corchetes
do
echo "opcion= $opcion"
echo -n "Ingrese un número [0 para salir]: "
read opcion
done       # esto finaliza el ciclo while
```

Este programa entra en un ciclo en el que pide el ingreso de un valor infinitamente hasta que se ingrese el 0. Nótese que en la condición del *while* los valores están separados de los corchetes y de los operadores.
Otro ciclo repetitivo muy utilizado en la programación es el ciclo **for**. Éste permite ejecutar un bloque de funciones una determinada cantidad de veces. Esta cantidad puede estar definida por el contenido de una variable. El funcionamiento de *for* en Bash puede parecerles raro; miren el siguiente ejemplo:

```
#!/bin/sh
# Ejemplo 5: uso del for.
valores=`ls`
for x in $valores
do
echo "El valor de x es: $x"
done
```

La variable **valores** contiene la salida del comando **ls** (listado de archivos). Esto se logra asignando a la variable el nombre del comando entre comillas invertidas. El **for** almacena en la variable **x** el valor que está siendo ciclado dentro de la variable **$valores**. **For** realiza un ciclo por cada valor dentro de la variable **$valores** (en el ejemplo, uno por cada archivo); por lo tanto, si tenemos diez archivos en un directorio, **for** realizará diez repeticiones del código.
Es posible, también, utilizar **for** del siguiente modo:

```
#!/bin/sh
# Ejemplo 5: uso del for.
for x in 1 2 3 4 5
do
echo "el valor de x es: $x"
done
```

De este modo, **for** realizará cinco ciclos del código especificado entre las sentencias *do* y *done*.

Creación de menús

Bash provee una función para la creación rápida y sencilla de menús con múltiples opciones: la función **select**. Su forma básica es la siguiente:

```
select [variable] in opcion1 opcion2 opcion3 opcion4 (...)
do
   sentencias
done
```

Así, **select** creará un menú numerado del 0 en adelante con cada opción definida en la función. Mostrará todas las opciones en pantalla, y luego aparecerá el símbolo **#?**, el cual esperará el ingreso del número de opción.
Luego, en **[variable]** quedará guardada la opción elegida; de este modo es posible definir las diferentes acciones con funciones **if**. Para aclarar un poco las cosas, veamos el siguiente ejemplo:

¡ADOPTEN EL USO DE FUNCIONES!

El uso de funciones hace al buen programador. Las funciones son la base de la programación estructurada. Les permitirán obtener un código reusable, más prolijo y fácil de actualizar.

EL SISTEMA DE MENÚS DE BASH

Mediante el uso de **select**, Bash nos provee de una sencilla manera de crear menús de selección para nuestros programas.

```
10: select opcion in "Hola Mundo"  "Adiós mundo cruel"
20: do
30: if [" $opcion = "Hola Mundo" ]
40: then
50: echo "Hola Mundo"
60: elif [ $opcion = "Adiós mundo cruel" ]
70: echo "Adiós mundo cruel"
80: break
90: fi
100: done
```

En la línea 10 se definen las opciones disponibles. Éstas, si están encerradas entre comillas, pueden contener varias palabras. Entre las líneas 30 y 90 se definen las diferentes acciones para las dos opciones. El ciclo es cortado por la función **break**.

Creación de funciones personalizadas

Podemos crear nuestras propias funciones en Bash. Ésta es su forma básica:

```
mi_funcion() {
 sentencias(...)
}
```

Cabe destacar que estas funciones personalizadas deben ser incluidas al principio del código, y no al final, como en otros lenguajes.
Si queremos realizar un llamado a la función, solamente incluimos el nombre y los parámetros del siguiente modo:

```
mi_funcion param1 param2 (...)
```

Dentro de la función creada, los parámetros son llamados igual que en todos los scripts Bash (**$1**, **$2**, etc.). Para devolver un valor numérico al término de una función, se utiliza la sentencia **return**. Su forma es:

```
return [valor]
```

El valor es definido por el usuario.

Ejecución de comandos

Pueden incluir comandos completos entre las líneas de su script Bash. Éstos serán ejecutados como si fueran lanzados desde la línea de comandos. También pueden utilizar la función **exec** para ejecutar un programa. Su forma de utilización es la siguiente:

```
exec [programa]
```

Aquí, **[programa]** puede ser una variable con la cadena por ejecutar. Cabe destacar que el uso de **exec** de esta manera mata el proceso padre; en nuestro caso, el script.

Terminando el programa

Podemos terminar un ciclo repetitivo con la función **break**, o el programa completo con la función **exit**. Esta última puede devolver un valor para saber si un script hijo ha terminado de manera correcta o no. Su forma es:

```
exit [valor]
```

En resumen...

El lenguaje de programación que incluye el intérprete Bash es una herramienta muy interesante a la hora de realizar programas que automaticen tareas. Si bien es un poco más críptico que los otros lenguajes de script, permite hacer muchas cosas en pocas líneas. No está de más aprender a trabajar con él.

LA BIBLIA DE

Capítulo 5

La programación en lenguaje Perl

Perl es uno de los lenguajes más famosos de estos tiempos. Su facilidad de uso y su velocidad de procesamiento lo convierten en el más apto para el desarrollo de aplicaciones CGI para la Web.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

El lenguaje Perl

Generalmente, Perl se incluye por defecto en la instalación de la mayoría de las distribuciones. Para asegurarnos de esto, podemos buscar el archivo **/usr/bin/perl**. Si no lo tienen, pueden instalarlo directamente del CD que acompaña a este libro. Una vez instalado, tipeen **perl -v**. El programa devolverá:

```
This is perl, v5.6.0 built for darwin

Copyright 1987-2000, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5.0 source kit.

Complete documentation for Perl, including FAQ lists, should be found on this
system using `man perl' or `perldoc perl'.  If you have access to the
Internet, point your browser at http://www.perl.com/, the Perl Home Page.
```

Funcionamiento general

Perl fue especialmente diseñado para el manejo de bases de datos textuales. Permite el manejo de variables, arreglos, estructuras condicionales y bucles. Para crear un script en Perl, sigan los mismos pasos que para crear uno en Bash o en Tcl. La única diferencia radica en la ruta del intérprete.
A continuación, un pequeño script en Perl. Para crearlo, pueden utilizar el editor VI (recuerden luego otorgarle permisos de ejecución).

```
#!/usr/bin/perl
# Ejemplo: ¡Hola Mundo!

print "¡Hola Mundo!";
```

La ruta al intérprete del lenguaje Perl es **/usr/bin/perl**. Luego de un comentario, figura la funcion **print** con la cadena "¡Hola Mundo!" como parámetro.

Variables

Tal como los otros lenguajes que hemos aprendido hasta el momento, Perl permite el manejo de variables de manera sumamente sencilla. Las variables se definen del siguiente modo:

```
$variable = valor;
```

Nótese el símbolo "$" antes del nombre de la variable. Esto define que es una variable escalar. Luego se ingresa el valor, que puede ser un entero, un flotante o una bien una cadena.
Para imprimir una variable, pueden utilizar la función **print** del siguiente modo:

```
print "El valor de la variable 'variable' es: $variable"
```

Por último, cabe destacar que cada función se finaliza con un símbolo ";", tal como en el lenguaje de programación C.

Arreglos de variables

Los arreglos de variables son formas de agrupar variables en un mismo conjunto. Son fáciles de definir; la manera más sencilla es la siguiente:

```
@arreglo = (val1, val2, val3, ...);
```

A diferencia de las variables, los arreglos comienzan con un simbolo "@".
Para imprimir un valor de un arreglo se utiliza la función **print** del siguiente modo:

```
print "El valor 2 del arreglo es: @arreglo[2]";
```

Si quieren imprimir el arreglo completo, pueden utilizar la función **print** con el siguiente contenido:

```
print "El arreglo completo contiene: @arreglo"
```

Arreglos con etiquetas

Una particularidad de Perl es la posibilidad de crear arreglos con etiquetas para cada valor. Para crear un arreglo de este tipo, se utiliza el simbolo "%" antes del nombre.

```
%mi_arreglo("valor1", 10, "valor2", 20, "valor3", 30);
```

El arreglo *"mi arreglo"* contiene tres valores que pueden ser llamados mediante las etiquetas *valor1*, *valor2* y *valor3*.

```
print "El valor 1 es: $mi_arreglo{valor1}"
```

Noten que la llamada a la etiqueta del arreglo se realiza con las llaves. El siguiente es un ejemplo de arreglos con etiquetas:

Ejemplo de arreglos con etiquetas

```
#!/usr/bin/perl
# Ejemplo: Arreglos con etiquetas

%familia=("papa", 47, "mama", 45, "hermano", 15, "hermana", 12);
print "Padre: $familia{papa}";
print "Madre: $familia{papa}";
print "Hermano: $familia{papa}";
print "Hermana: $familia{papa}";
```

UNA VARIABLE ESPECIAL:

Perl tiene definida una variable llamada "$_", la cual contiene los datos ingresados por la entrada estándar.

LOS ARREGLOS ETIQUETADOS

Este modo de almacenar la información en memoria también se llama "hash". Lo encontrarán con este nombre en numerosos libros y artículos de revistas.

En este ejemplo se define un arreglo con las etiquetas de los integrantes de una familia y sus respectivas edades. Finalmente, se las imprime utilizando la función **print**.

Expresiones

Perl permite evaluar expresiones de forma sencilla. Veamos el siguiente ejemplo:

```
10:  #!/usr/bin/perl
20:  # Ejemplo: Expresiones
30:  $val1=2;
40:  $val2=5;
50:  print "Realizando Multiplicacion...";
60:  $val3=$val1*$val2;
70:  print "$val3";
```

En este pequeño programa definimos dos variables (líneas 30 y 40) con los valores 2 y 5. Se imprime un pequeño mensaje y se define una tercera variable con el resultado de la multiplicación de las dos anteriores. Por último, se imprime la última variable en pantalla.

Perl ofrece otros comandos de evaluación de expresiones muy útiles:

eq	Comparar dos cadenas
ne	Diferencia
lt	Menor que
gt	Mayor que

Para concatenar dos cadenas, se puede utilizar el operador de punto ".". El siguiente es un ejemplo:

```
$nombre = "Carlos";
$apellido = " Lopez";
$nombre_completo = $nombre.$apellido
```

Por consiguiente, si imprimimos la variable **$nombre_completo**, obtendremos como resultado "Carlos Lopez".

Ingreso de datos

El ingreso de datos en Perl es muy sencillo. Para realizar esta acción, se define una variable con el flujo de entrada estándar como ingreso. El siguiente es un ejemplo:

```
$entrada=<STDIN>
```

Recuerden que el flujo de entrada estándar es **STDIN**. Estas definiciones pueden aparecer en cualquier parte del código.

```
#!/usr/bin/perl

print "Ingrese su edad:";
$edad = <STDIN>;
print ("Su edad es: $edad");
```

Este pequeño programa muestra un mensaje y luego define una variable para realizar el ingreso de datos. Por último, se lo muestra en pantalla como una variable cualquiera. Recuerden siempre utilizar la función **chop** para eliminar el carácter de retorno de carro en las cadenas. Su forma es:

```
chop ($variable=<STDIN>);
```

La estructura "if"

La estructura if en Perl es muy similar a la del lenguaje C. La forma básica es:

```
if ( condicion ) {
    funciones
 }

else  {
    funciones
}
```

Como pueden ver, la condición se encierra entre paréntesis, y las funciones respectivas a cada opción se encierran entre llaves. Veamos un ejemplo práctico:

Ejemplo de estructuras condicionales

```
#!/usr/bin/perl
print "Ingrese un numero:";
$numero=<STDIN>;
if ($numero >= 1000) {
            print "Su número es mayor (o igual) que 1000!";
        }
 else {
        print "Su número es menor que 1000!";
 }
```

Los operadores que pueden utilizar en la condición son los siguientes:

eq	Igual a...	>=	Mayor o igual que
>	Mayor que	<=	Menor o igual que
<	Menor que	**ne**	Diferente

La estructura "unless"

Ésta es otra de las particularidades de Perl. La estructura **unless** ejecuta una serie de funciones sólo si una condición es falsa. Veamos el siguiente ejemplo:

```
unless (numero > 1000) {
      print "Su numero es menor que 1000!";
}
```

La función "while"

Perl también puede manejar bucles condicionales como **while**. La estructura de **while** en Perl es la siguiente:

```
while (condicion)
{
funciones..."
}
```

También pueden usar la entrada estándar para recibir los datos utilizando **while**. Su forma es la siguiente:

```
while(<STDIN>) {
 funciones...
}
```

Ciclo "for"

Como en todo lenguaje de programación, el ciclo **for** nos permite repetir una porción de código una determinada cantidad de veces. Esa cantidad de veces se define mediante tres expresiones. Su forma es la siguiente:

```
for (expresion1; expresion2; expresion3) {
      funciones...
}
```

Como en los lenguajes anteriormente mencionados, pueden omitir cualquiera de las expresiones, pero deben incluir los puntos y comas. El siguiente es un simple ejemplo: un contador del uno al diez.

```
#!/usr/bin/perl

for($a=0 ; $a <= 10; $a++ ) {
print "\n$a";
}
```

Escritura de datos en archivos

Perl fue especialmente diseñado para trabajar con datos en archivos de texto. Ofrece muchísimas herramientas que permiten almacenar datos, leer y realizar búsquedas sobre archivos de texto. Incluso, muchos programadores logran armar complejas bases de datos trabajando sobre archivos de texto. Si bien su estudio completo lleva mucho más que un par de páginas de un libro, en esta sección intentaremos dar una visión general de cómo trabaja el sistema. Perl utiliza la función **open** para realizar la apertura de archivos. Su forma es la siguiente:

```
open (NOMBRE, "archivo");
```

El nombre es el descriptor del archivo en cuestión; el archivo se encierra siempre entre comillas. Puede incluir la ruta completa a un archivo, si es que éste se encuentra en otro directorio que no sea el de ejecución.
Por defecto, Perl abre los archivos en modo de sólo lectura. Para abrirlo en modo escritura, se agrega un símbolo ">" al principio de la ruta al archivo, tal como el siguiente:

```
open(NOMBRE, ">archivo");
```

Si se quiere abrir el archivo para modo "append" (agregar los datos al final), se agregan dos símbolos ">":

```
open(NOMBRE, ">>archivo");
```

Una vez abierto el archivo, ya podemos escribir sobre él. Como ejemplo, se puede utilizar la función **print** del siguiente modo:

```
print NOMBRE "cadena"
```

Esta función imprimirá la cadena en el archivo identificado con la etiqueta "**NOMBRE**". Veamos el siguiente ejemplo:

Ejemplo de uso de archivos

```
10:  #!/usr/bin/perl
20:  open (ARCHIVO, ">>file.txt");
30:  print "Ingrese su nombre: ";
40:  chop($nombre=<STDIN>);
50:  print "Ingrese su apellido: ";
60:  chop($apellido=<STDIN>);
70:  print ARCHIVO "$apellido, $nombre\n";
80:  close ARCHIVO;
```

En este ejemplo se abre un archivo llamado **file.txt** para modo *append* (línea 20). En las líneas 40 y 60 se hace el ingreso de datos utilizando la función **chop**, para que elimine el carácter final de la cadena (de esta manera, los datos se imprimirán en el archivo uno al lado del otro). Por último, en la línea 70 se imprime en **ARCHIVO** la cadena formateada, y en la línea 80 se cierra el archivo.

Lectura de archivos

Para la lectura de archivos, la utilización de funciones (generales) es la misma. Primero se debe abrir el archivo del siguiente modo:

```
open (ARCHIVO, "file.txt");
```

Luego, pueden almacenarse en una variable los datos del archivo, siempre línea por línea, del siguiente modo:

```
#!/usr/bin/perl

open(ARCHIVO, "file.txt");
$linea=<ARCHIVO>;
print $linea;
```

También pueden utilizar una función **while** para procesar todas las líneas sin tener que repetir código. Una forma puede ser la siguiente:

```
#!/usr/bin/perl

open(ARCHIVO, "file.txt");

while (<ARCHIVO>) {
print $_;
}

close ARCHIVO;
```


Cómo definir subfunciones

Perl posibilita la creación de subfunciones para permitir al programador manejarse en un confortable entorno de programación estructurada.
Las funciones en Perl se definen al principio del código, utilizando la función **sub** del siguiente modo:

```
sub mi_funcion {
  (...)
}
```

Luego, para llamar a la función creada, directamente utilizamos su nombre: **mi_funcion;**.
También es posible pasar parámetros a las funciones. Por defecto, estos parámetros se guardan en "@_" dentro de la función. Veamos el siguiente ejemplo:

```
10:  #!/usr/bin/perl
20:  sub imprime {
30:  print "\nEl contenido de los parametros es @_";
40:  }

50:  imprime "Hola Mundo";
```

Este programa crea una función en la línea 20 que imprime el contenido de la variable "@ ". En la línea 50 se realiza el llamado a la función, pasándole como parámetro la cadena "Hola Mundo".

Ejecución de programas

Para ejecutar programas, Perl ofrece diferentes métodos. Uno de ellos es mediante la función "system", que ejecuta directamente como comando la cadena que se le pasa como parámetro. Su forma es la siguiente:

```
system "ls"
```

También pueden utilizar las comillas invertidas (``) encerrando un comando para guardar su salida en una variable. Un ejemplo puede ser el siguiente:

```
$salida=`ls -l`
```

De este modo, si imprimimos la variable "salida", tendremos como resultado el listado del directorio actual.

Bases de datos en Perl

DBM es un sistema de bases de datos que ya lleva muchos años en el campo de los sistemas operativos UNIX. Si bien es un sistema extremadamente sencillo en todos sus aspectos, es muy útil cuando se quiere desarrollar pequeñas aplicaciones que no hagan uso intensivo de funciones complejas de manejo de datos. Muchos programadores lo utilizan para desarrollar pequeños sistemas de almacenamiento de datos que no vayan a ser ejecutados en grandes equipos (en términos de hardware) y requieran sencillez y velocidad.
Actualmente, existen varias implementaciones del UNIX DBM (gDBM del proyecto GNU, nDBM y oDBM son algunos ejemplos de ellas). Perl utiliza su propia implementación (sDBM), que si bien no es tan potente como la gDBM, posee casi todas las funciones fundamentales. De todas formas, se supone que todas estas implementaciones son compatibles, por lo que si algún día quieren migrar a otro sistema de DBM, no tendrán que modificar el código. Obviamente, son sólo suposiciones.
Una cosa realmente importante para destacar es que usar el sistema DBM en Perl nos asegura que nuestro sistema funcionará en cualquier computadora que tenga un intérprete de Perl instalado, aun si se trata de diferentes sistemas operativos. De esta forma, un sistema que usa DBM desarrollado en Perl para GNU/Linux funciona correctamente y sin modificaciones de código en un sistema que posee Mac OS X o Windows.

¿Cómo funciona DBM?

En Perl, todo el control de bases de datos DBM se realiza por medio de los ya conocidos *hashes*. Nuestro querido comando **tie**, con algunos parámetros extra, nos permitirá crear, abrir, modificar y cerrar bases de datos. Estos parámetros extra son los siguientes:

- El nombre del hash.
- El nombre del módulo DBM.
- El archivo de la base de datos.
- Definición de opciones de acceso al archivo.
- Permisos que tendrá el archivo.

Una función que sirve de ejemplo para crear una base de datos es la siguiente:

```
tie %dbm, 'SDBM File', "base.db", O_CREAT O_RDWR, 0644;
```

Como pueden ver, el uso de esta función es realmente sencillo. El primer parámetro, **%dbm**, nos servirá para hacer referencia a nuestra base de datos. El archivo de base de datos lo definimos en el tercer parámetro (**base.db**), y haremos que Perl lo cree con permisos de lectura y de escritura para nosotros (**O_CREAT|O_RDWR, 0644**). Una aclaración: el cuarto parámetro define la forma en que Perl accederá al archivo. El quinto parámetro indica los permisos que el archivo tendrá una vez creado.

Revisión de una base de datos

Y aquí también podemos hacer uso de la sentencia **or die**, en caso de que la base de datos no haya sido creada. Por ejemplo:

```
tie %dbm, 'SDBM_File', "archivo.db", O_RDWR, 0 or die "Error abriendo
archivo.db\n";
```

EL SISTEMA DE BASE DE DATOS DE PER.

El sistema de base de datos de Perl, aunque resulte bastante precario para muchos experimentados con "monstruos" como MySQL o PostgreSQL (no llega ni a los talones de estos dos), es muy útil para desarrollar pequeñas aplicaciones y hacer manejos rápidos de datos. Además, es sumamente portable, ya que donde haya un intérprete de Perl, seguramente estarán las librerías para manejo de DBM.

Este recurso es muy sencillo de implementar, y nos será de gran utilidad cuando llegue la hora de realizar el proceso de detección de errores en el código.

Ingreso de datos

Las bases de datos DBM, al usar el sistema de hashes, funcionan bajo el modelo de "palabra_clave=valor". De esta forma, usando la sintaxis de hashes en Perl, para agregar un nuevo dato a nuestra base de datos tendremos que tipear algo como lo siguiente:

```
$dbm{'nombre'}="Facundo";
```

Ahora bien, ¿qué sucede si queremos manejar múltiples registros? Bueno, tendremos que ingeniárnoslas para arreglarnos con este sistema. Una buena idea puede ser implementar la base de datos bajo la siguiente estructura:

```
1.nombre="Facundo"
1.apellido="Arena"
2.nombre="Fernando"
2.apellido="Perez"
```

Y así sucesivamente. El primer número es útil para asignar un valor de registro a cada una de las personas que tengamos en nuestra base de datos.

Acceso a datos

Para acceder a alguno de los datos de nuestra base de datos, la cosa es bastante similar. Por ejemplo, podemos asignar un registro a una variable del siguiente modo:

EL ACCESO A DATOS

Como pueden apreciar, el acceso a datos albergados en una base de datos DBM de Perl se maneja exactamente igual que un hash (un arreglo etiquetado), ya que la interfaz de datos que estamos usando para manejarnos con la base son, justamente, los arreglos etiquetados. El mismo concepto se aplica para el ingreso de datos. En conclusión, no hay mucho nuevo que aprender. ¿Quién diría que es tan sencillo?

```
$variable=%dbm{'1.nombre'};
```

Siguiendo la línea de nuestros ejemplos, **$variable** contendría la palabra "Facundo". De todas formas, pueden realizar un acceso directo al hash. El siguiente es un ejemplo para mostrar el contenido de una palabra clave en pantalla:

```
print "El contenido del primer registro es %dbm{'1.nombre'};
```

Es recomendable que antes de intentar acceder a alguna de las palabras clave de nuestra base de datos, realicemos una revisión, para cerciorarnos de que ese valor realmente existe. Una de las formas de hacer esto es encerrando el acceso a la palabra clave dentro de una función **if**, del siguiente modo:

```
if(exists $dbm{'palabraclave'}) {

[...]

}
```

Por último, para borrar una entrada en la base de datos, tienen que usar la función **delete** del siguiente modo:

```
delete %dbm{'palabraclave'};
```

Cerrando una base de datos

Y para seguir con la línea de las cosas sencillas, diremos que cerrar una base de datos es cuestión de usar la función **untie** seguida del nombre de la base de datos. El siguiente es un ejemplo válido de esto:

LA FUNCION EXISTS

La función exists es muy útil, ya que devuelve un valor verdadero cuando existe un elemento del hash definido. Esta función, combinada con la estructura if, nos permite realizar rutinas que hagan una revisión para que no se permita definir un elemento de hash que ya haya sido definido previamente.

```
untie %dbm;
```

Recuerden que deben cerrar la base de datos una vez terminado el acceso a ella. Esta operación se realiza generalmente al finalizar la ejecución del programa.

En resumen...

El lenguaje de programación Perl goza de enorme popularidad entre los usuarios de GNU/Linux (y los de otros sistemas operativos también, ya que es multiplataforma) gracias a su facilidad de uso, su velocidad de procesamiento y su estabilidad. Actualmente, es muy utilizado para desarrollar un tipo de aplicaciones denominadas CGI, que son las que funcionan por detrás de los servidores web (por ejemplo, un buscador, un foro de mensajes, etc.).
En este capítulo, hemos visto de manera general cómo es la estructura del lenguaje, así como el modo de manejarnos con su sistema interno de base de datos. Si les interesa CGI, hay muchísima documentación en Internet sobre este tema, y no tendrán problemas para entenderlo, ya que no harán más que aplicar de manera combinada los conocimientos que obtendrán con este libro.

LA BIBLIA DE

Capítulo 6

El modo gráfico

Vamos a dejar de teclear un poco para internarnos en el mundo gráfico de GNU/Linux.

SERVICIO DE ATENCIÓN AL LECTOR lectores@tectimes.com

El sistema Xwindow

Hasta ahora nos hemos manejado con el sistema por medio de una interfaz textual, tipeando comandos y visualizando mensajes de color blanco sobre una aburrida pantalla de fondo negro. Este *modo texto* es el que se viene utilizando desde la aparición de las primeras pantallas para computadoras, y aún hoy es el preferido de muchos usuarios.
Actualmente, el modo texto es muy usado por los expertos administradores de sistemas UNIX, ya que es una forma mucho más flexible de operar el sistema. Ahora bien, estamos en el tercer milenio, la tecnología gráfica de las computadoras ha avanzado enormemente y hay ciertas funciones que son mucho más sencillas de realizar desde una interfaz gráfica. Aquí es donde aparece Xwindow.
Xwindow es un entorno gráfico originalmente diseñado en el MIT (*Masachussets Institute of Technology*) para que fuera parte de todos los sistemas UNIX que existían en esa época. Pero Xwindow es mucho más que una simple interfaz gráfica. Funciona bajo el modelo de sistema cliente/servidor, y hasta posee un protocolo interno de comunicación para las aplicaciones. ¿Qué quiere decir esto? Sencillo: es posible correr un *servidor X* en una computadora y varios clientes en diferentes terminales de la red. Estos clientes pueden ejecutar aplicaciones que estén alojadas en el servidor, sin necesidad de procesamiento en el sistema local. En todo sistema UNIX con Xwindow, existe una variable de entorno llamada **$DISPLAY** que indica la dirección IP del servidor X, seguida de un símbolo ":" y un número (generalmente, el 0) que indica con qué servidor X se quiere establecer una comunicación.
Ahora bien, en GNU/Linux, el entorno gráfico posee el nombre de **Xfree86**. Éste es una implementación libre del clásico Xwindow. Incluso es totalmente compatible con él. De ahora en más, cada vez que hagamos referencia al entorno gráfico de Linux, lo llamaremos Xfree86.
Por último, mencionaremos que el usuario puede interactuar con el entorno gráfico mediante el uso de la pantalla (monitor), el mouse y el teclado. Todos estos componentes se configuran mediante la herramienta **xf86config**, sobre la cual trataremos en detalle más adelante

Administradores de ventanas

Como todo en el mundo de GNU/Linux, lo que el usuario ve como un solo componente en pantalla, en realidad, son varios subcomponentes más pequeños. El administrador de ventanas es una aplicación cliente que se encarga del manejo de las ventanas correspondientes a las aplicaciones del usuario. ¿Redundante? Vamos a simplificarlo un poco. El administrador de ventanas se encarga de ofrecer al usuario las herramientas básicas para que pueda maximizar una ventana, minimizarla, cambiar su posición, esconderla, cambiar su tamaño, etc.

Existen varios administradores de ventanas, de entre los cuales el usuario puede elegir el que más le agrade. La diferencia básica entre todos ellos radica en el aspecto visual, el rendimiento y la cantidad de funciones que ofrece. Hagamos un breve repaso de los más famosos.

- **FVWM.** éste es uno de los primeros administradores de ventanas que aparecieron para GNU/Linux. Durante muchos años, era el sistema que el usuario veía por defecto en pantalla luego de instalar su distribución de GNU/Linux favorita. FVWM es un sistema muy completo y sumamente flexible. Permite minimizar las ventanas, cambiar su tamaño, posee un menú flotante de aplicaciones y mucho más. Existe una versión especial de FVWM llamada FVWM95, la cual es una versión configurada de FVWM para que luzca exactamente igual al entorno de Windows 95. Es interesante si se va a instalar un sistema a una persona que aún no tiene muchos conocimientos en entornos gráficos.
  Si bien ha quedado en un segundo plano por el uso de los administradores de escritorio FVWM es ideal en muchos casos. El primero de ellos es cuando necesitamos usar el entorno gráfico en un sistema con pocos recursos de hardware. FVWM consume muy poco y ofrece mucho. En sistemas medianamente actuales, FVWM es muy veloz y permitirá que la carga de aplicaciones se realice mucho más rápido.
  También muchos "gurúes" del mundo de GNU/Linux y UNIX lo eligen como administrador de ventanas preferido, por la cantidad de funciones que ofrece, su velocidad y su flexibilidad. Podrán encontrar una versión de FVWM en el CD de su distribución favorita, o pueden descargarlo de **www.fvwm.org**.

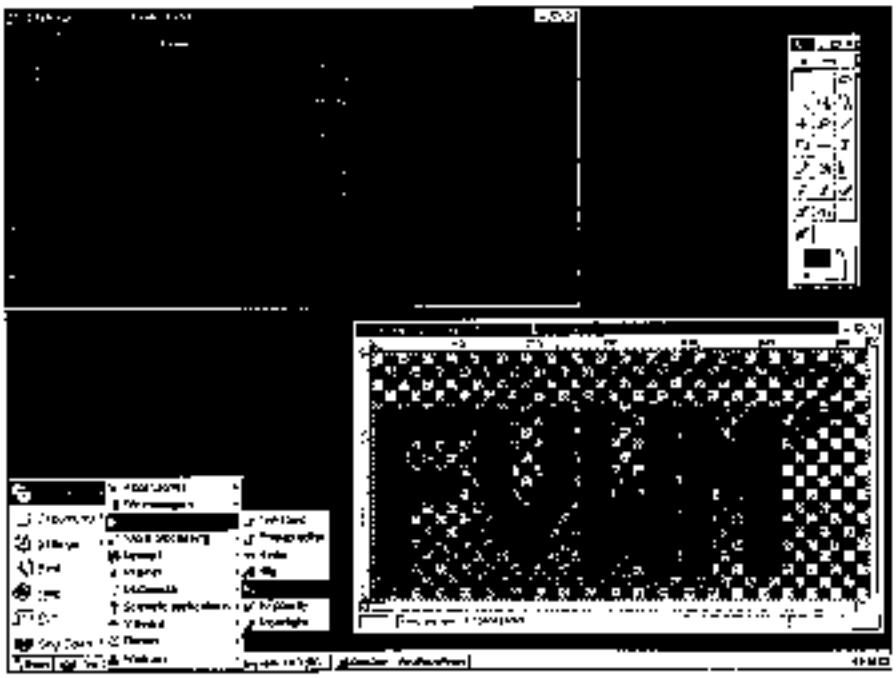


Figura 1. Aunque antiguo, FVWM puede ser totalmente configurado para lucir como el más avanzado de los administradores de ventanas.

- **WindowMaker**: originario del Brasil, WindowMaker es uno de los proyectos de Software Libre para administrador de ventanas más activos. Su aspecto visual es similar al de Afterstep, y permite realizar muchas funciones sobre las ventanas. Posee un sistema de minimización basado en "docks" o pequeños cubículos que se ubican en la parte inferior de la pantalla y corresponden a las diferentes aplicaciones minimizadas. Su aspecto visual es realmente de avanzada, todos los iconos están desarrollados con gran detalle, y es posible configurar el sistema para cambiar su aspecto totalmente. Cabe destacar que, si bien es un sistema muy completo y muy flexible, consume algo más de recursos que sus competidores.

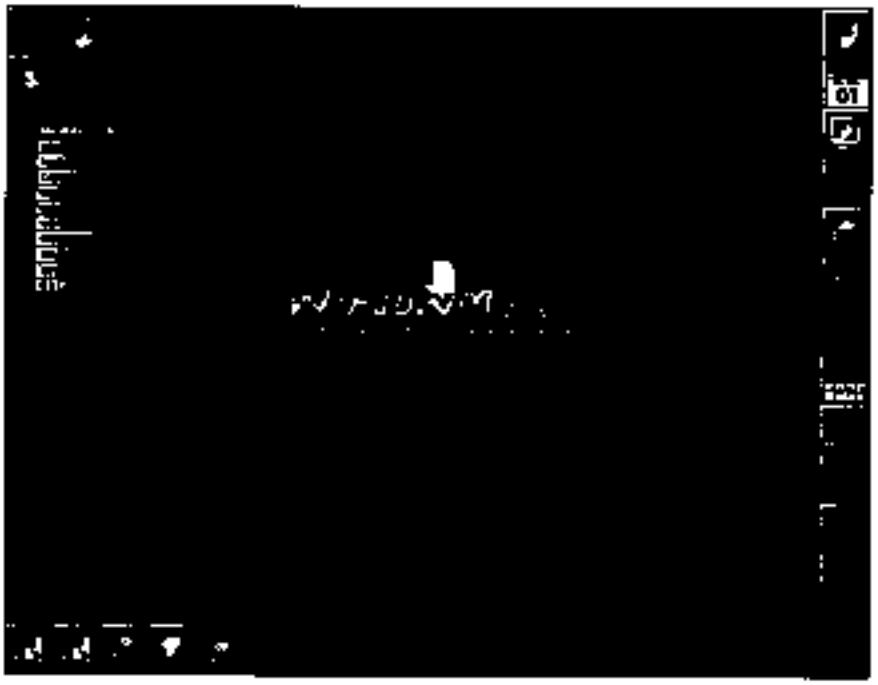


Figura 2. WindowMaker, el administrador de ventanas elegido por millones de usuarios de GNU/Linux.

- **Afterstep**: este administrador de ventanas es casi tan viejo como el mismísimo FVWM. Afterstep surge de la idea de imitar el entorno gráfico del sistema Next (NextStep), y lo hace realmente bien. Aunque actualmente su desarrollo no es tan intensivo, estamos hablando de un sistema que ya es un clásico en los diferentes entornos UNIX y posee muchos adeptos. En funciones, es muy similar a WindowMaker;

CLIENTES X

Existen clientes de Xwindow para casi todas las plataformas. Esto quiere decir que una persona que usa, por ejemplo, Windows XP puede conectarse a un servidor X de GNU/Linux mediante la red y ejecutar las aplicaciones de forma totalmente transparente.

es posible tener un menú flotante que nos despliega un listado de aplicaciones separadas en categorías y podemos escoger de allí la que queremos ejecutar.

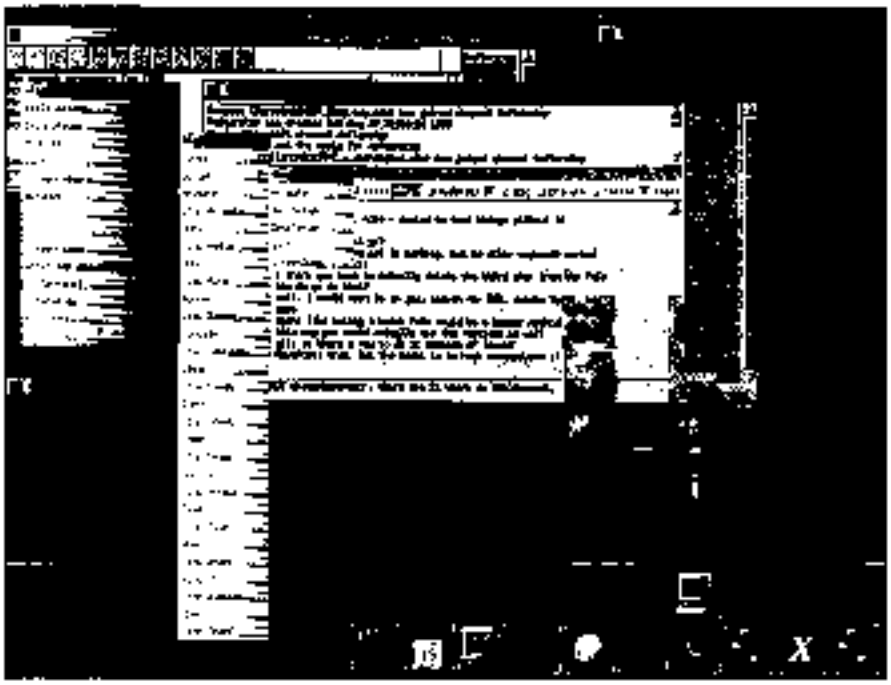


Figura 3. Como pueden observar, el look visual de Afterstep es bastante sofisticado. Además, es totalmente configurable.

- **Oroborus**: el proyecto Oroborus tiene como principal finalidad desarrollar un administrador de ventanas pequeño pero sumamente funcional (generalmente ocupa 50k, y se van a asombrar con la cantidad de posibilidades que ofrece). Al ser tan pequeño, no brinda algunas herramientas que otros administradores de ventanas incluyen. Por esta razón, se pueden utilizar ciertas aplicaciones como **KeyLaunch** (para configurar combinaciones de teclas), **deskmenu** (un menú principal) y algún panel (como el de GNOME, por ejemplo). Todas estas aplicaciones pueden ser descargadas directamente desde el sitio de Oroborus (**www.dreamind.de/oroborus. shtml**).
  En el ámbito de Mac OS X, Oroborus es muy conocido por su versión OroborOSX (**http://oroborosx.sourceforge.net/**) , que actúa como administrador de ventanas en el sistema Xwindow para Mac OS X y tiene un look similar a Acqua (la interfaz de OS X). De esta forma, las aplicaciones de Mac OS X y las de Xwindow pueden coexistir en un mismo entorno, y sus ventanas se ven exactamente igual.
  Si bien hay muchísimos administradores de ventanas para GNU/Linux, Oroborus puede ser una opción interesante para tener en cuenta a la hora de utilizar un administrador de ventanas que consuma poca memoria y funcione velozmente.

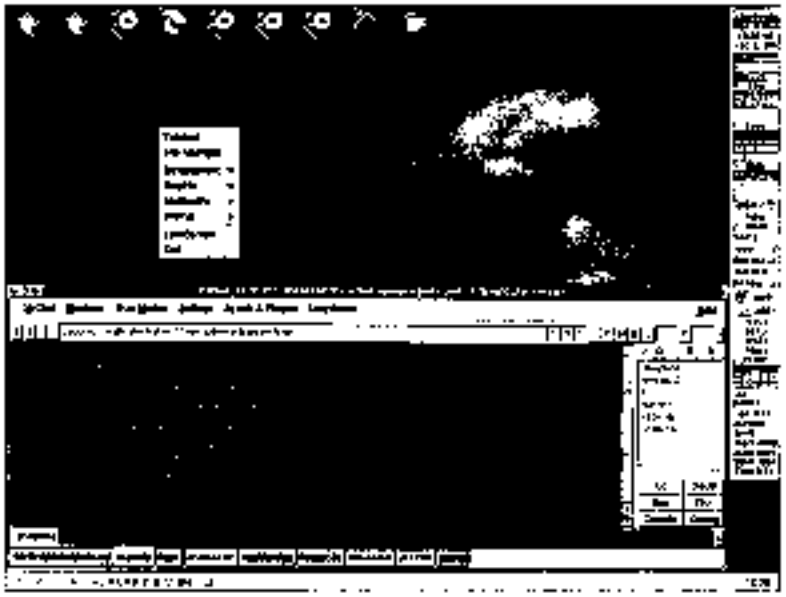


Figura 4 Aquí podemos ver a Oroborus en acción. En la imagen, está funcionando el administrador de ventanas con un entorno visual modificado.

- **Blackbox**: éste es un administrador de ventanas relativamente nuevo en el mundo de GNU/Linux, y se caracteriza principalmente por su velocidad y por su poca consumición de recursos. Con su aparición, Blackbox ha permitido que los usuarios de viejas computadoras con procesadores 486 puedan ejecutar XfreeB6 con un administrador de ventanas más que aceptable y su aplicación preferida. Si bien no es muy completo y tampoco es muy flexible, es la única opción cuando se quiere instalar un entorno gráfico en sistemas con pocos recursos.

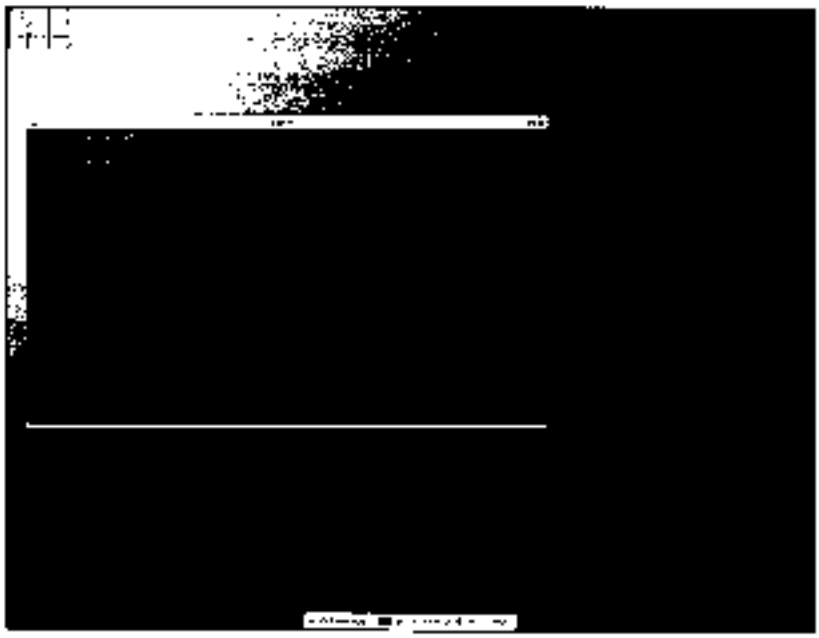


Figura 5. BlackBox. El administrador de ventanas que permite usar un entorno gráfico en sistemas con muy bajos recursos.

Administradores de escritorios

Pronto, las necesidades de los usuarios fueron cada vez mayores y los administradores de ventanas tuvieron que evolucionar en lo que ahora llamamos *administradores de escritorios*. Éstos, básicamente, poseen todas las funciones de un administrador de ventanas, pero además nos proveen de una barra inferior con un menú de aplicaciones, botones correspondientes a las aplicaciones abiertas, varios escritorios virtuales de trabajo, iconos permanentes con accesos directos a unidades, aplicaciones o directorios, y un set de herramientas como un editor de textos, un visualizador de imágenes, un navegador web y mucho más. Actualmente hay dos administradores de escritorios que dominan el mundo de GNU/Linux. Veamos cuáles son:

- **KDE**: éste fue el primer administrador de escritorios que apareció para GNU/Linux, y quizá ésta sea la razón por la cual tiene tantos adeptos. KDE es un completo entorno visual que, además de realizar el manejo de ventanas y de múltiples escritorios, posee un completo set de herramientas de altísima calidad para el usuario. Para Internet, KDE ofrece Konqueror, un excelente navegador web que se integra en el sistema para funcionar también como administrador de archivos local. Además, incluye un cliente de e-mail, un sistema de mensajería instantánea y un entorno de configuración de una cuenta PPP.

Figura 6. KDE en pleno funcionamiento. Como pueden observar, el entorno está sumamente detallado a nivel gráfico y es altamente configurable.

- **GNOME:** en alguna época, KDE no era totalmente Software Libre, porque trabajaba sobre unas librerías que estaban regidas bajo una licencia que no se adecuaba al concepto Software Libre. Entonces, en el proyecto GNU decidieron desarrollar un administrador de escritorios totalmente libre. Así fue como nació GNOME, un entorno

totalmente Software Libre desarrollado sobre las librerías GTK (las utilizadas para crear el famoso sistema de retoque fotográfico The GIMP). Más tarde, el proyecto KDE pasó a ser un proyecto totalmente Software Libre, y ambos quedaron en el mismo nivel. A grandes rasgos, ambos ofrecen las mismas posibilidades para el usuario, aunque, como siempre, en el ámbito linuxero hay fanáticos de uno y otro bando.

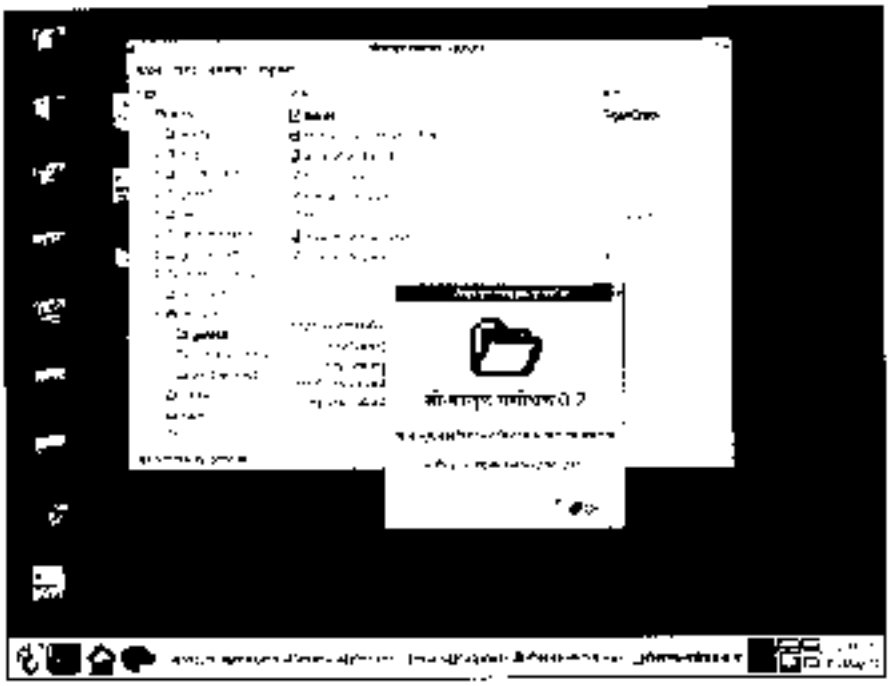


Figura 7. Y aquí tenemos a GNOME, el otro administrador de escritorios. Como pueden notar, los componentes del escritorio son realmente similares a los de KDE.

Aplicaciones gráficas contra aplicaciones en modo texto

Habiendo explicado esto, queda entonces claro que en el sistema operativo GNU/Linux podemos trabajar bajo dos entornos diferentes: el modo texto y el modo gráfico.

EL SITIO OFICIAL DE GNOME

El proyecto GNOME posee su propio sitio oficial en la dirección www.gnome.org.

EL SITIO OFICIAL DE KDE

Y el proyecto KDE también tiene el suyo, puedes encontrarlo en www.kde.org.

Ahora bien, ¿cuál es la más conveniente? Bueno, eso depende de la utilidad que se le vaya a dar al sistema operativo.
El modo texto es más adecuado cuando se realizan trabajos "serios", como administración de redes, desarrollo de aplicaciones, monitoreo de sistemas, etc. El modo texto ofrece varias ventajas ante el modo gráfico: la independencia de un entorno gráfico para funcionar, velocidad, infinidad de opciones y parámetros por programa, mayor estabilidad, etc. Pero a los usuarios hogareños y de oficina, el modo texto les resulta complicado y aburrido. Por eso hacen uso intensivo del modo gráfico.
Y las aplicaciones de modo gráfico no siempre son tan productivas como las del modo texto. Generalmente, son más lentas y propensas a cuelgues. Pero la decisión queda en ustedes. Ojo, en muchos casos, ocurre que una aplicación de modo gráfico ofrece más funcionalidades que su contraparte en modo texto, y viceversa. Para estas ocasiones, saber dominar ambos entornos es lo más práctico.
En este libro explicaremos cómo realizar las tareas más importantes desde el modo texto, ya que si se comprende el funcionamiento en este entorno, el modo gráfico será mucho más fácil de entender. Pongamos manos a la obra, y comencemos con la configuración del entorno gráfico, desde el modo texto.

Configuración usando xf86config

X11 posee un sistema de configuración de sus dispositivos primordiales (mouse, teclado, monitor y placa de video), llamado xf86config. Esta herramienta está incluida en todas las distribuciones que contengan el paquete oficial Xfree86. Pero si usan otra variante de UNIX, también contarán con él. Para ejecutarlo, tipeen desde el *shell* la siguiente línea de comando:

```
xf86config
```

El programa mostrará un pequeño mensaje de bienvenida, y luego de presionar la tecla **ENTER**, se dará lugar a la configuración del sistema.

LA CONFIGURACION DE XWINDOW

La forma de configurar Xwindow que utilizaremos en este libro es considerada la más complicada, por ser muy poco amigable y en modo texto. De todas formas, esta herramienta es la única que se encuentra en absolutamente todas las distribuciones de GNU/Linux que incluyan Xfree86, por lo que será necesario que aprendan a dominarla. Tengan mucho cuidado cuando comiencen a hacer sus primeras pruebas.

```
First specify a mouse protocol type. Choose one from the following list:

 1.  Microsoft compatible (2-button protocol)
 2.  Mouse Systems (3-button protocol)
 3.  Bus Mouse
 4.  PS/2 Mouse
 5.  Logitech Mouse (serial, old type, Logitech protocol)
 6.  Logitech MouseMan (Microsoft compatible)
 7.  MM Series
 8.  MM HitTablet
 9.  Microsoft IntelliMouse
10.  Acecad tablet

If you have a two-button mouse, it is most likely of type 1, and if you have
a three-button mouse, it can probably support both protocol 1 and 2. There
are two main varieties of the latter type: mice with a switch to select the
protocol, and mice that default to 1 and require a button to be held at
boot-time to select protocol 2. Some mice can be convinced to do 2 by sending
a special sequence to the serial port (see the ClearDTR/ClearRTS options).

Enter a protocol number:
```

Esta pantalla les permitirá seleccionar un protocolo para el mouse; generalmente se utilizan mouses del tipo *"Microsoft Compatible"* (opción 1), pero si ustedes están completamente seguros de tener otro modelo, elíjanlo.

```
You have selected a Microsoft protocol mouse. If your mouse was made by
Logitech, you might want to enable ChordMiddle which could cause the third
button to work.

Please answer the following question with either 'y' or 'n'.
Do you want to enable ChordMiddle?
```

Aquí, el programa les da la posibilidad de activar el ChordMiddle. Este sistema permite activar el tercer botón del mouse. En Linux, el botón del medio tiene múltiples utilidades. Respondan a esta opción con "y".

```
You have selected a three-button mouse protocol. It is recommended that you
do not enable Emulate3Buttons, unless the third button doesn't work.

Please answer the following question with either 'y' or 'n'.
Do you want to enable Emulate3Buttons?
```

Ésta es otra opción de emulación de tres botones para el mouse. Si luego de toda la configuración el sistema no reconoce el tercer botón, habrá que realizar el proceso de configuración nuevamente activando esta opción. Cabe destacar que no pueden estar los dos sistemas de emulación de tres botones activados. Si activan ChordMiddle, no pueden activar Emulate3Buttons, y viceversa. Por ahora, responderemos a esta pregunta con "n".

```
Now give the full device name that the mouse is connected to, for example
/dev/ttyS0. Just pressing enter will use the default, /dev/mouse.

Mouse device:
```

Ahora, el programa nos pregunta la dirección del dispositivo de mouse. Tal como figura en la tabla del **Capítulo 2**, el dispositivo del mouse se encuentra en **/dev/mouse**. Sólo presionen **ENTER** en esta pregunta, y el programa lo configurará automáticamente.

```
Beginning with XFree86 3.1.2D, you can use the new X11R6.1 XKEYBOARD
extension to manage the keyboard layout. If you answer 'n' to the following
question, the server will use the old method, and you have to adjust your
keyboard layout with xmodmap.

Please answer the following question with either 'y' or 'n'.
Do you want to use XKB?
```

Este sistema nos permite configurar el tipo de teclado. En este ejemplo vamos a suponer que tenemos instalado un teclado en español. Respondamos "**y**" a esta pregunta. A continuación, el programa nos mostrará un pequeño listado de las distribuciones de teclas soportadas:

```
List of preconfigured keymaps:

  1  Standard 101-key, US encoding
  2  Microsoft Natural, US encoding
  3  KeyTronic FlexPro, US encoding
  4  Standard 101-key, US encoding with ISO9995-3 extensions
  5  Standard 101-key, German encoding
  6  Standard 101-key, French encoding
  7  Standard 101-key, Thai encoding
  8  Standard 101-key, Swiss/German encoding
  9  Standard 101-key, Swiss/French encoding
 10  Standard 101-key, US international
 11  Brazilian ABNT2
 12  None of the above

Enter a number to choose the keymap.
```

Como podemos apreciar, no se encuentra el teclado en español en esa lista. Por esta razón, elegimos la opción 12: "Ninguna de ésas" (*None of the above*).

```
You did not select one of the preconfigured keymaps. We will now try to
compose a suitable XKB setting. This setting is untested.
Please select one of the following standard keyboards. Use DEFAULT if
nothing really fits (101-key, tune manually)

  1  Standard 101-key keyboard
  2  Standard 102-key keyboard
  3  101-key with ALT_R = Multi_key
  4  102-key with ALT_R = Multi_key
  5  Microsoft Natural keyboard
  6  KeyTronic FlexPro keyboard
  7  DEFAULT

Enter a number to choose the keyboard.
```

El sistema intentará ahora ser un poco más específico con los detalles del teclado, y nos mostrará una pantalla de selección de teclado estándar. Aquí podemos elegir la opción 2: "Teclado 102-teclas estándar".

Luego de un mensaje, el programa nos pedirá que elijamos uno de los siguientes países del listado:

```
1  Belgium
2  Brazil/ABNT2 Layout
3  Bulgaria
4  Canada
5  Czechoslovakia
6  Denmark
7  Finland
8  France
9  Germany
10  Italy
11  Norway
12  Poland
13  Portugal
14  Russia
15  Spain
16  Sweden
17  Thailand
18  Switzerland/French layout

Enter a number to choose the country.
Press enter for the next page
```

Ya que el único país de habla española que figura en la lista es España, lo elegimos presionando la opción 15.
Ya está concluida la configuración de nuestro teclado en español. Ahora, xf86config pasará a otra etapa de configuración: la del monitor. Para esto nos mostrará una serie de configuraciones estándar:

Y SI SE EQUIVOCAN...

No tienen más que apretar la combinación de teclas CTRL+C para eliminar el proceso xf86config. No corren ningún riesgo, ya que el archivo se modifica luego de finalizado el proceso de configuración.

```
It is VERY IMPORTANT that you do not specify a monitor type with a horizontal
sync range that is beyond the capabilities of your monitor. If in doubt,
choose a conservative setting.

    hsync in kHz; monitor type with characteristic modes
 1  31.5; Standard VGA, 640x480 @ 60 Hz
 2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
 3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
 4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
 5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
 6  31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
 7  31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
 8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
 9  31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
10  31.5 - 95.0; Monitor that can do 1280x1024 @ 85 Hz
11  Enter your own horizontal sync range
```

Si poseemos una placa de video SuperVGA, podemos elegir la opción 2. Igualmente, no es recomendable elegir una de estas opciones si tenemos los manuales originales del monitor. Si los tienen, pueden poner su propia configuración de sincronización horizontal de pantalla (opción 11), y obtendrán mejores resultados. En nuestro ejemplo, elegiremos la opción 2.

```
You must indicate the vertical sync range of your monitor. You can either
select one of the predefined ranges below that correspond to
industry-standard monitor types, or give a specific range. For interlaced
modes, the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).

 1  50-70
 2  50-90
 3  50-100
 4  40-150
 5  Enter your own vertical sync range

Enter your choice:
```

Ahora nos pide el rango de barrido horizontal de la pantalla. Podemos elegir uno de los estándar o poner el nuestro propio (eligiendo la opción 5). Para continuar, elegimos la opción 2.

El programa luego nos pedirá que ingresemos unas etiquetas para hacer referencia al monitor en el archivo de configuración. En el caso del ejemplo, el monitor es un SyncMaster 3. Pueden elegir el nombre que ustedes deseen; esta sección no alterará el funcionamiento del programa.

```
The strings are free-form, spaces are allowed.
Enter an identifier for your monitor definition: Samsung
Enter the vendor name of your monitor: Samsung
Enter the model name of your monitor: SyncMaster 3
```

Aquí viene una de las secciones más importantes del programa: la elección de la placa de video. Si queremos ver un listado, deberemos responder "y" cuando lo pregunte.

```
1     3DLabs Oxygen GMX                    PERMEDIA 2
2     3DVision-1740 AGP                    Intel 740
(...)
10    AOpen PA2010                         Voodoo Banshee

(...)
```

Esta lista es mucho más extensa, demasiado como para agregarla a este libro.
Si quieren elegir alguno de los servidores genéricos, pueden responder "N" a la última pregunta. Esto hará que xf86config les muestre el siguiente listado de servidores:

```
1  The XF86_Mono server. This a monochrome server that should work on any
VGA-compatible card, in 640x480 (more on some SVGA chipsets).
2  The XF86_VGA16 server. This is a 16-color VGA server that should work on
any VGA-compatible card.
3  The XF86_SVGA server. This is a 256 color SVGA server that supports
a number of SVGA chipsets. On some chipsets it is accelerated or supports
higher color depths.
4  The accelerated servers. These include XF86_S3, XF86_Mach32, XF86_Mach8,
XF86_8514, XF86_P9000, XF86_AGX, XF86_W32, XF86_Mach64, XF86_I128 and
    XF86_S3V.

These four server types correspond to the four different "Screen" sections
in XF86Config (vga2, vga16, svga, accel).

Which one of these screen types do you intend to run by default (1-4)?
```

El servidor **XF86_Mono** es sólo para monitores VGA compatibles. Únicamente muestra dos colores. Si tienen problemas con los otros servidores, pueden probar con éste.
El servidor **XF86_VGA16** es el más compatible con las placas VGA y SVGA. Soporta sólo 16 colores.
El servidor **XF86_SVGA** no es compatible con todas las placas SVGA, pero soporta 256 colores.
Los demás son servidores acelerados para placas más recientes.

En este ejemplo, utilizaremos el servidor XF86_SVGA, eligiendo la **opción 2**. Luego, el programa preguntará:

```
The server to run is selected by changing the symbolic link 'X'. For
example,'rm /usr/X11R6/bin/X; ln -s /usr/X11R6/bin/XF86_SVGA
/usr/X11R6/bin/X' selects
the SVGA server.

The directory /var/X11R6/bin exists. On many Linux systems this is the
preferred location of the symbolic link 'X'. You can select this
location when setting the symbolic link.

Please answer the following question with either 'y' or 'n'.
Do you want me to set the symbolic link?
Do you want to set it in /var/X11R6/bin?
```

Aquí, el sistema pregunta si quieren crear un link simbólico a la placa seleccionada. Respondan "y" a las dos preguntas.
Luego, el programa pedirá que elijan la cantidad de memoria de video que posee su placa. Las opciones estándar que posee *xf86config* son las siguientes:

```
How much video memory do you have on your video card:

  1  256K
  2  512K
  3  1024K
  4  2048K
  5  4096K
  6  Other

Enter your choice:
```

Si su placa posee más de cuatro megas de memoria, pueden elegir la opción 6 e ingresar la cantidad en KBs.
Luego, el sistema pedirá el ingreso de etiquetas para la placa de video, que, al igual que las etiquetas identificatorias del monitor, serán útiles dentro del archivo **xf86config**.

```
The strings are free-form, spaces are allowed.
Enter an identifier for your video card definition: Mi placa de video You can
simply press enter here if you have a generic card, or want to describe your
card with one string.
Enter the vendor name of your video card: SVGA Genarica
Enter the model (board) name of your video card: SVGA
```

El proceso de configuración va llegando a su fin. Una de las últimas secciones es la de selección del *ClockChip* de su placa de video.

```
A Clockchip line in the Device section forces the detection of a
programmable clock device. With a clockchip enabled, any required
clock can be programmed without requiring probing of clocks or a
Clocks line. Most cards don't have a programmable clock chip.
Choose from the following list:

 1  Chrontel 8391                                                  ch8391
 2  ICD2061A and compatibles (ICS9161A, DCS2824)                   icd2061a
 3  ICS2595                                                        ics2595
 4  ICS5342 (similar to SDAC, but not completely compatible)       ics5342
 5  ICS5341                                                        ics5341
 6  S3 GenDAC (86C708) and ICS5300 (autodetected)                  s3gendac
 7  S3 SDAC (86C716)                                               s3_sdac
 8  STG 1703 (autodetected)                                        stg1703
 9  Sierra SC11412                                                 sc11412
10  TI 3025 (autodetected)                                         ti3025
11  TI 3026 (autodetected)                                         ti3026
12  IBM RGB 51x/52x (autodetected)                                 ibm_rgb5xx

Just press enter if you don't want a Clockchip setting.
What Clockchip setting do you want (1-12)?
```

Si no están seguros de qué opción elegir, sólo presionen **ENTER**. El programa les dará la posibilidad de ejecutar un *"autoprobe"* para detectar el ClockChip exacto.

```
For most modern configurations, a Clocks line is neither required or
desirable.  However for some older hardware it can be useful since it
prevents the slow and nasty sounding clock probing at server start-up.
Probed clocks are displayed at server startup, along with other server and
hardware configuration info. You can save this information in a file by
running 'X -probeonly 2>output_file'. Be warned that clock probing is
inherently imprecise; some clocks may be slightly too high (varies per run).

At this point I can run X -probeonly, and try to extract the clock
information from the output. It is recommended that you do this yourself and
if a set of clocks is shown then you add a clocks line (note that the list
of clocks may be split over multiple Clocks lines) to your Device section
afterwards. Be aware that a clocks line is not appropriate for most modern
hardware that has programmable clocks.

You must be root to be able to run X -probeonly now.

Do you want me to run 'X -probeonly' now?
```

Éste es un momento crucial de la configuración: si responden "y" y la placa de video está configurada correctamente, el programa de configuración automáticamente detectará los modos de pantalla correspondientes: todo concluirá con éxito
Si la placa de video no está correctamente configurada o no está soportada por su versión de XFree, entonces en esta etapa **xf86config** se congelará o devolverá un mensaje de error. En este caso, habrá que realizar toda la configuración otra vez.
Si responden "n", se utilizarán los valores estándar (no es recomendable).
Para finalizar, **xf86config** les informa de los modos de pantalla en los que el servidor X podrá ejecutarse:

DÓNDE SE GUARDA TODO

Toda la configuración se guardará en el archivo /etc/X11/XF86Config. Éste es un archivo de texto, y podrá ser modificado con cualquier editor de textos.

```
For each depth, a list of modes (resolutions) is defined. The default
resolution that the server will start-up with will be the first listed
mode that can be supported by the monitor and card.
Currently it is set to:

"640x480" "800x600" "1024x768" "1280x1024" for 8bpp
"640x480" "800x600" "1024x768" "1280x1024" for 16bpp
"640x480" "800x600" "1024x768" "1280x1024" for 24bpp
"640x480" "800x600" "1024x768" for 32bpp

Note that 16, 24 and 32bpp are only supported on a few configurations.
Modes that cannot be supported due to monitor or clock constraints will
be automatically skipped by the server.

 1  Change the modes for 8pp (256 colors)
 2  Change the modes for 16bpp (32K/64K colors)
 3  Change the modes for 24bpp (24-bit color, packed pixel)
 4  Change the modes for 32bpp (24-bit color)
 5  The modes are OK, continue.

Enter your choice:
```

Estos datos son generalmente correctos; podemos oprimir la **opción 5**. De todos modos, si ustedes quieren modificar alguna resolución, pueden hacerlo eligiendo el modo (8,16, 24 o 32) e ingresando la resolución deseada.
Por último, **xf86config** preguntará si quieren guardar la nueva configuración en el archivo **/etc/X11/xf86config**. Si creen que está todo correctamente, presionen **"y"** y el programa concluirá su ejecución.

Cómo cambiar de administrador de ventanas/escritorios

El sistema Xfree86 revisa un archivo ubicado en el directorio **$HOME**, llamado **.xclients** (noten el punto inicial, que indica que es un archivo oculto). En ese archivo, se indica por medio de una función **exec** (es un simple script de Bash) cuál es el

comando que se ejecutará una vez cargado el entorno gráfico. Allí es donde debe figurar el comando que inicia su gestor de ventanas/escritorios actual. Por ejemplo, en un sistema que ejecuta KDE, deberá haber una línea similar a la siguiente:

```
exec kde
```

Si quisiéramos cambiar a GNOME:

```
exec gnome-session
```

Ahora es sólo cuestión de averiguar cuál es el ejecutable del administrador de ventanas que se quiere usar, y simplemente configurarlo en este archivo.

En resumen...

El sistema Xfree86 es la implementación en código abierto del famoso Xwindow. Si bien su configuración puede ser muy sencilla en algunas distribuciones, en otras puede no serlo tanto. Esto, básicamente, dependerá de qué herramientas de configuración incluya cada distribución. En este libro, aprendimos a configurarlo casi manualmente, usando las herramientas estándar que encontraremos en todas las distribuciones de GNU/Linux que incluyan Xfree86.
Además, aprendimos qué es un administrador de ventanas y cuáles son las diferencias con los administradores de escritorios. Por último, vimos cómo configurar nuestro gestor de ventanas o de escritorios preferido para que sea cargado por defecto cada vez que ingresamos en el modo gráfico.

LA BIBLIA DE

Capítulo 7

GNU/Linux en red

Antes de comenzar a ver los diferentes servicios de red, vamos a configurar nuestros sistemas. Luego haremos uso de algunos servicios del protocolo TCP/IP.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

En red...

Para montar una red en Linux, o en cualquier otro sistema operativo, primero hay que definir las necesidades y los objetivos. Un servidor con procesamiento en paralelo y 256 MB de RAM puede ser un poder excesivo para manejar cuatro computadoras, y un Pentium 133 puede ser demasiado poco para manejar 15 sistemas simultáneamente. Por esta razón, lo primero que tienen que hacer es sentarse y definir ciertos puntos básicos:

1. Cuántas computadoras van a estar conectadas a la red.
2. Cuántos usuarios tendrán acceso simultáneo a la red.
3. Qué tipo de programas correrán en la red.
4. Cuántos servidores son necesarios.
5. ¿Es necesario dedicar un servidor especial para la base de datos?

Todas estas preguntas llevan tiempo e investigación. Es necesario que los administradores, ustedes, averigüen los requerimientos de hardware del software que van a usar en la red y los servicios que van a proveer. Todo esto se obtiene leyendo la documentación de los programas.
Trabajaremos sobre una idea imaginaria de instalar una red hogareña de cuatro máquinas con los servicios esenciales de red. Luego, la escalabilidad será sencilla una vez que tengan los conceptos básicos bien claros.
Para montar una red se necesitan, básicamente, tres componentes: computadoras, cables y un concentrador que se encargue de interconectar las computadoras. Las computadoras deben tener instalada una placa de red, que es la encargada de recibir la información y enviarla por el cable para que las otras computadoras puedan recibirla. Se pueden encontrar muchos modelos y marcas de placas de red, a diferentes precios. Una vez que instalamos las placas de red en las computadoras, ya podemos conectarlas al Hub (concentrador) mediante cables UTP. Conectar las computadoras al Hub es realmente sencillo, simplemente hay que enchufar una de las puntas del cable UTP a la computadora, y la otra, a cualquiera de las entradas del Hub.
Ahora que tenemos todo instalado, es momento de que elijamos una distribución de Linux y comencemos a configurar el sistema de red.

SENDMAIL

La mayor parte de las distribuciones de GNU/Linux utilizan Sendmail como agente de transporte de e-mails. Este programa es ejecutado al inicio del sistema por el script /etc/rc.d/init.d/sendmail.

El software

En este capítulo veremos cómo configurar los servicios básicos de cualquier red montada sobre un sistema operativo Linux. Por esta razón, cualquier distribución reconocida será apta para manejar nuestra red. A modo de sugerencia, recomendamos instalar alguna de las distribuciones más "serias" y renombradas en el ámbito de los servidores, como Slackware o Debian. Para las estaciones de trabajo, quizá sea recomendable instalar una distribución más amigable, como Red Hat, SuSE (incluida en esta edición) o Mandrake, ya que las herramientas proporcionadas deben ser fácilmente dominadas por el operador de la estación. De todas formas, el proceso de configuración no cambiará en absoluto, ya que configuraremos los archivos a mano y con un editor de texto convencional (como emacs o VI). De esta manera, obtendrán mayor experiencia, y después podrán buscar una herramienta de automatización que se adecue a sus necesidades.

Configuración de la placa de red

Lo primero que debe hacerse en todas las máquinas de la red (incluido el servidor) es configurar el dispositivo de red. Esto se lleva a cabo mediante el comando **modprobe**, el cual se encarga de levantar un dispositivo desde el nivel de usuario. El formato básico del comando **modprobe** es el siguiente:

```
modprobe [dispositivo]
```

El parámetro **[dispositivo]** es el nombre del módulo que maneja el dispositivo por cargar. Por ejemplo, para cargar una placa de red **RealTek 8139** (las más conocidas y baratas del mercado) sólo tienen que tipear el comando **modprobe rtl8139**.
Ahora bien, es esencial que sepan el modelo de placa de red que instalaron en la computadora. De otra forma, encontrar el nombre del módulo será un trabajo muy duro. Pueden buscar el nombre del módulo en la documentación del código fuente del kernel, o directamente en el directorio **/lib/modules/kernel/drivers**.
Ya tenemos la placa de red cargada y lista para ser usada, pero sólo por esta vez. Si apagan la computadora y vuelven a encenderla, tendrán que cargar el módulo nuevamente. Para evitar esto, es recomendable editar el archivo **/etc/rc.d/rc.local** y agregar como última línea el comando **modprobe** completo para cargar la placa de red. De ahora en más, cada vez que el sistema se inicie ejecutará el módulo de la placa de red.

Asignando direcciones IP

Para que nuestras computadoras puedan reconocerse entre ellas, cada una tendrá asignada una dirección. Estas direcciones son conocidas como direcciones **IP**
No vamos a explicar cuál es la estructura y la forma de las direcciones **IP**, pero trabajaremos de forma esencial con ellas para que nuestra red funcione correctamente.
El administrador debe asignar a cada computadora una dirección **IP** que consta de cuatro números entre 0 y 255, separados por puntos.
Pueden elegir cualquiera de los números disponibles, menos la dirección **127.0.0.1**, ya que ésta es la dirección estándar que apunta a la misma máquina.
Supongamos que nuestra red tiene cuatro máquinas. Una configuración recomendable sería usar la dirección **10.0.0.1** para el servidor **10.0.0.2** para la primera estación, **10.0.0.3** para la segunda estación, y **10.0.0.4** para la tercera estación.
Más adelante veremos cómo asignar nombres a cada una de nuestras computadoras, de forma tal que no sea necesario andar recordando las direcciones IP de cada máquina, sino que bastará con identificarlas por su denominación.
Cuando cargamos el módulo controlador de nuestra placa de red, **modprobe** creó un dispositivo llamado **eth0** Recuerden esto cada vez que necesiten hacer referencia a la placa de red en algún programa.
Para asignar una dirección **IP** a una máquina, se utiliza el comando **ifconfig** de la siguiente forma:

```
ifconfig eth0 up 10.0.0.1
```

El primer parámetro le indica a **ifconfig** que debe trabajar con el dispositivo **eth0** (nuestra placa de red).
El segundo parámetro le indica a **ifconfig** que debe cargar una dirección IP, indicada en el tercer parámetro.
Para verificar que todo está configurado correctamente, tipearemos el comando **ifconfig** sin parámetros. La salida indicará el dispositivo **lo0** (**loopback**) y el dispositivo **eth0** con información detallada de éste. Si no aparece el dispositivo **eth0**, entonces tendrán que asegurarse que la placa de red fue correctamente instalada y cargada por el comando **modprobe**.
Repitan el mismo paso en cada máquina, cambiando solamente el último dígito de la dirección IP, tal como lo explicamos antes.
¡Pero cuidado! Tengamos en cuenta nuevamente que los datos se borrarán si apagamos el sistema, por lo que es recomendable agregar también la línea del **ifconfig** a nuestro archivo **/etc/rc.d/rc.local**

Verificando la conexión

Antes de comenzar a configurar algún servicio para nuestra red, vamos a verificar que la conexión esté correctamente realizada y que las computadoras puedan *hablar* entre sí. Para esto, nada mejor que el conocido comando **ping**. La función del comando **ping** consiste en enviar un paquete de datos a una dirección IP especificada y esperar a que esta máquina le responda. Si no responde durante un periodo de tiempo, el comando devolverá un error, y será necesario verificar la conexión y la configuración de las placas. La forma básica del comando **ping** es:

```
ping [dirección ip]
```

Por eso, si queremos comprobar desde el servidor la conexión con la estación 1, sólo tendremos que tipear **ping 10.0.0.2**. El sistema comenzará a mostrar en pantalla el resultado de la operación. Si entra en un ciclo, pueden terminarlo con la combinación de teclas **CTRL+C**.

Asignando nombres de host

Hasta este punto, cada vez que tengamos que hacer referencia a una máquina, tendremos que hacerlo mediante su dirección IP (por ej.: 10.0.0.2). Y si queremos hacer referencia a un usuario, tendremos que usar algo como **usuario@10.0.0.3**. Ante esta situación tenemos dos opciones. La primera de ellas consiste en instalar un servidor de DNS, algo que nos llevaría mucho tiempo, esfuerzo y páginas de este libro.
La segunda opción consiste en crear un archivo **/etc/hosts** genérico para todas las computadoras. En este archivo se listan las direcciones IP de todas las máquinas y sus correspondientes nombres. Por ejemplo, nuestro archivo **/etc/hosts** podría contener la siguiente información:

```
10.0.0.1        red.servidor
10.0.0.2        red.máquina1
10.0.0.3        red.máquina2
10.0.0.4        red.máquina3
```

Abran el archivo **/etc/hosts** con su editor favorito e ingresen la información que más se adecue a la red que tienen instalada. De ahora en más, en lugar de hacer referencia a **usuario1** usando la notación **usuario1@10.0.0.3**, podremos hacerlo mediante

usuario1@red.máquina1. Este sistema de nombres nos permitirá crear una estructura fácil de mantener y de recordar.
Si en un futuro planean incorporar más computadoras a la red, entonces instalar un servidor de nombres (DNS) será una tarea para tener seriamente en cuenta.
Ahora que ya tenemos nuestra red configurada, podemos comenzar a configurar los diferentes servicios de red.

Usando servicios de TCP/IP

En esta sección del libro aprovecharemos que ya tenemos nuestro dispositivo de red configurado, y haremos uso de los protocolos más difundidos en Internet por medio de las herramientas que nos provee la mayor parte de las distribuciones de GNU/Linux.

El sistema de documentos hipertextuales (http)

Este protocolo (puerto 80) es uno de los más utilizados para la transferencia de información por Internet. Consta de miles de servidores web y millones de documentos, todos conectados a la gran red.
El lenguaje en el que están escritos los documentos de la Web es el HTML (*HyperText Markup Language*), y éstos se transfieren mediante el protocolo HTTP (*HyperText Transfer Protocol*). Los clientes de estos archivos son programas especiales que sólo pueden interpretar el lenguaje HTML, y se llaman web browsers.
Los llamados documentos HTML pueden contener en su interior links o enlaces a otras páginas web. Esto nos permite crear un documento organizado en secciones y conectarlo o otra página web.
Para ser identificada, cada página de Internet tiene una URL asignada (*Uniform Resource Locator*). El formato de la URL es el siguiente:

```
http://www.alternarama.com.ar:80/index.html
```

El primer campo (**http://**) es el tipo de protocolo que se utilizará para acceder a ese documento. El segundo campo, **www.arena.org.ar**, es el dominio de esa página (un nombre fácil de recordar). El tercer campo, **index.html**, es el archivo al cual se quiere acceder.
Hay varios tipos de protocolos en TCP/IP; cada uno de ellos puede ir al principio de la URL:

file:// Acceder a un archivo en la máquina local.
ftp:// Acceder a un servidor FTP remoto.

gopher://	Acceder a un servidor gopher.
http://	Acceder a un servidor web.
mailto:	Enviar un e-mail a la dirección especificada.
news://	Acceder a un servidor de noticias (newsgroup).
telnet://	Acceder a un sistema remoto.
wais://	Acceder a un servidor wais.

Ahora que ya entendemos el funcionamiento general de la World Wide Web, tenemos que acceder a ella desde nuestra máquina Linux.

- **Mozilla:** éste es uno de los mejores navegadores para GNU/Linux. Está incluido en la mayoría de las distribuciones (Red Hat, Slackware, SuSE, etc.), y su utilización es bastante sencilla.
  Sus principales contras son la velocidad y la cantidad de recursos del sistema que consume, aunque si eso no es problema, entonces podrán disfrutarlo, ya que es muy estable. Pueden descargar la última versión de Mozilla desde **www.mozilla.org**.
  Mozilla incluye, además, un sistema lector de e-mail, un editor de páginas web, un cliente de IRC y un lector de noticias.

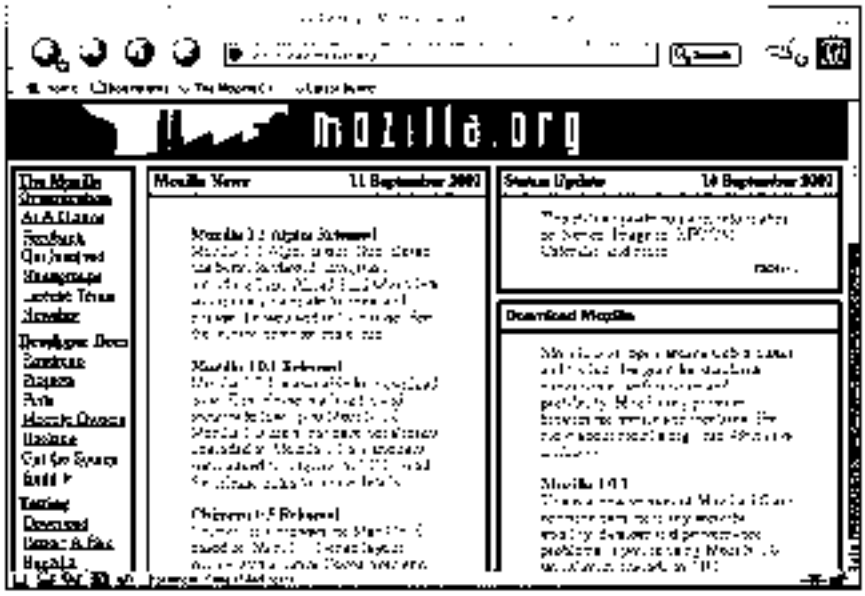


Figura 1. Mozilla, uno de los mejores navegadores para GNU/Linux.

- **Lynx**: si les molestan los gráficos en las páginas web o quieren mayor velocidad de transferencia de información, entonces necesitan un navegador en modo texto. El más famoso de éstos es *Lynx*, un completo navegador rápido y fácil de usar.
  No muestra gráficos (sólo sus etiquetas), ni tablas ni frames. Aun así, es uno de los navegadores más utilizados por los usuarios de Linux. Está incluido en la gran mayoría de las distribuciones, y se ejecuta tipeando "*lynx*" desde la línea de comandos. En la parte inferior de la pantalla, en encuentran las teclas de control del programa. Éstas son:

TECLA	FUNCIÓN
H	Nos envía a la pantalla de ayuda.
O	Muestra la pantalla de opciones, desde la cual podemos cambiar el editor por defecto, colores, tipo de documento, etc.
P	Nos permite imprimir el documento en una terminal, en un archivo o en un mail
G	Presionando esta tecla, podremos ingresar la URL de la página por descargar.
M	Desde cualquiera de las subsecciones de Lynx, volvemos a la página principal presionando esta tecla.
/	Con esta tecla se puede buscar una cadena determinada dentro de la página.
Q	Salir de Lynx.

Si bien Lynx no es muy "agradable a la vista", es una de las formas más eficientes de navegar la Web. Si le dan un poco de tiempo, seguro que se acostumbrarán rápidamente.

- **Links:** este navegador permite visualizar páginas web en modo texto con tablas y colores. Además, se controla íntegramente mediante menús descolgables que permiten navegar las opciones del programa con mucha facilidad. *Links* nos permite buscar palabras dentro del documento de forma eficiente y descargar archivos mientras navegamos en la Web. Sin duda, uno de los mejores navegadores hasta la fecha.

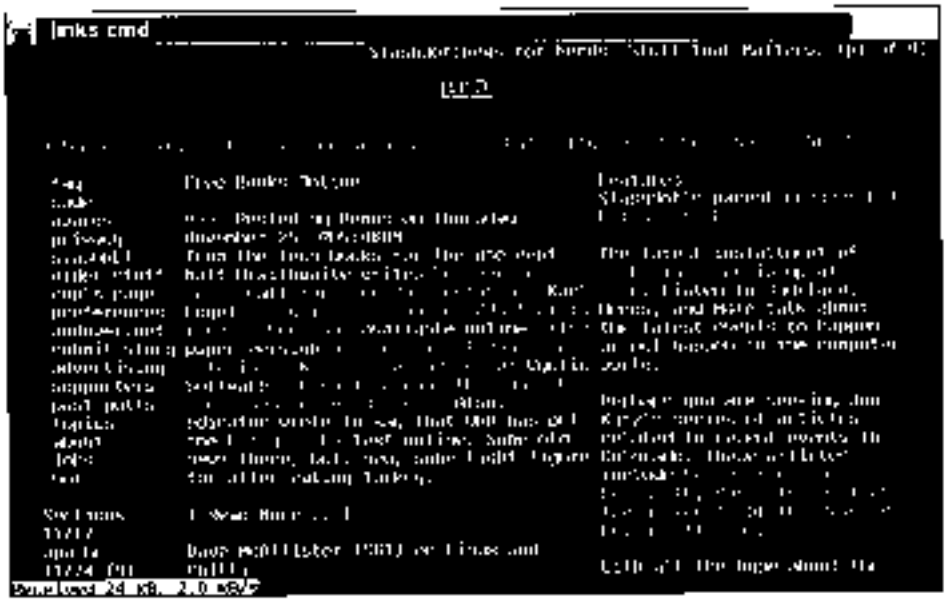


Figura 2. El navegador de modo texto Links visitando el sitio Slashdot.

Chat por IRC

Mediante este servicio de Internet se puede charlar con miles de personas de todo el mundo en tiempo real y por medio de mensajes. El IRC basa su orden en canales. Cada canal es creado por algún usuario.

Cuando un usuario crea un canal, le otorga un nombre y una descripción del tema de discusión en ese lugar. Hay cientos de miles de canales en IRC que tratan sobre los más diversos temas. Los usuarios no utilizan sus nombres verdaderos, sino que se asignan *"nicknames"* o apodos. Esto hace más divertida la sesión de chateo.
Los servidores IRC son aplicaciones que se ejecutan en máquinas conectadas a Internet y actúan como interconectores de todos los clientes. Para acceder a un servidor de éstos hay que tener un cliente de IRC y su respectiva dirección.
En Linux hay varios clientes de IRC; en esta sección detallaremos los más famosos. Pero antes de verlos, vamos a aprender los comandos de control de un servidor IRC que nos serán útiles y son comunes a todos los clientes.

COMANDO	DESCRIPCIÓN
/server direccion	Especificamos el servidor con el que nos conectaremos.
/join #canal	Entramos en el canal especificado. Recuerden que los nombres de los canales comienzan siempre con el simbolo #.
/nick nombre	El nombre de usuario o nickname que queremos utilizar en nuestra sesión IRC.
/list	Lista todos los canales disponibles en el servidor.
/quit	Cortar la sesión.

El protocolo NNTP

Éste es un sistema del que todo el mundo ha oido hablar, pero pocos saben de lo que realmente se trata.
Sus comienzos como protocolo de transmisión se remontan a la década del '80, cuando fue presentada su propuesta de estandarización para la transmisión de noticias.
Básicamente, es un sistema de mensajería. Los mensajes no van directamente a los usuarios, sino que quedan almacenados en el servidor. Si una persona quiere leer o escribir un mensaje, deberá conectarse al servidor y realizar la operación.
También pueden formarse hilos de mensajes. Esto quiere decir que se puede crear toda una discusión a partir de un solo mensaje, con sólo ir respondiendo a las opiniones de los diferentes miembros del grupo.
La mayoria de los proveedores de Internet otorgan un servidor de este tipo. De ahora en más, vamos a suponer que ustedes ya saben la dirección de su servidor de news.
Para acceder y operar sobre estos servidores se necesita un programa especial, comúnmente denominado News Reader (lector de noticias). Los más famosos dentro del mundo de Linux son **trn** y **tin**: **trn** es un derivado del viejo **rn** de UNIX. Estas aplicaciones generalmente se incluyen con las distribuciones de Linux, y pueden ejecutarlas tipeando su nombre directamente desde la línea de comandos. Los nombres de los grupos de noticias se organizan en diferentes clases. Un nombre de ejemplo podria ser:

```
comp.os.linux
```

Este nombre se divide en tres secciones separadas por puntos. La primera, "comp", indica que es un grupo de noticias en el que se habla de temas relacionados con la computación (hardware o software). La segunda, "os", indica que en este grupo se tratan temas exclusivos de sistemas operativos (os= operating systems).
Por último, la tercera sección, "*Linux*", indica que se habla solamente del sistema operativo GNU/Linux.
La primera sección del nombre del grupo indica la categoría general. La siguiente tabla lista todas las categorías posibles.

CATEGORÍA	DESCRIPCIÓN
alt	Grupo Alternativo. Se habla de temas genéricos.
bionet	Se discuten temas de biología.
bit	Se discuten temas de bitnet.
biz	Aquí se intercambian mensajes relacionados con los negocios.
clari	Noticias actualizadas día a día.
comp	Temas específicos de computación.
ieee	Exclusivo del IEEE (Institute of Electrical and Electronic Engineers).
k12	Dedicado a la educación primaria y secundaria.
misc	Se habla de temas variados.
news	Discusiones acerca de la administración de las news.
rec	Artes recreativas.
sci	Ciencia e Ingeniería.
soc	Sociedades y culturas.
talk	Discusiones de temas de actualidad.

Antes de comenzar a utilizar los lectores de noticias, es necesario preparar el sistema. Hay dos acciones básicas para realizar. La primera es definir una variable de entorno que contenga la dirección del servidor de news de nuestro proveedor. Para esto, simplemente tecleamos el siguiente mandato desde el Bash:

```
NNTPSERVER=news.suproveedor.com
export NNTPSERVER
```

La mayoría de los lectores de noticias utiliza esta variable.

El archivo .newsrc ubicado en el directorio **$HOME** sirve para definir configuraciones. Este archivo es útil para almacenar los grupos de noticias a los que accedemos

periódicamente. Simplemente creen este archivo en su directorio y complétenlo con los nombres de los grupos de noticias. Un ejemplo podría ser el siguiente:

```
comp.os.Linux.misc:
comp.os.Linux.setup:
```

Nótese que todas las líneas terminan con un símbolo ":". De ahora en más, cada vez que alguno de los programas de noticias vaya a efectuar la conexión con el servidor, realizará la petición de actualización de los mensajes de los grupos de noticias definidos en este archivo.
Una vez configurado el sistema, podremos comunicarnos con nuestro servidor de noticias haciendo uso de nuestro cliente favorito.

Correo electrónico

Uno de los servicios más utilizados de Internet (y tal vez, el más importante), es el correo electrónico (e-mail).
El sistema de funcionamiento es el siguiente: el usuario A escribe un mensaje al usuario B y lo envía. Este mensaje se almacena en un directorio especial, y cuando el usuario B se conecta al sistema, el programa de e-mail le informa que tiene un mensaje esperando a ser leído.
Tal vez pueda parecer que esta teoría de funcionamiento es solamente verdadera para una red de uso local, pero lo es también para una WAN (como Internet). La única diferencia radica en que, cuando se le envía el mensaje al usuario B, si éste se encuentra en otra LAN, hay que especificar la dirección (el dominio) de ésta.
En Linux, los mensajes se almacenan en el directorio **/var/spool/mail** . Dentro de ese directorio hay archivos con los mails de todos los usuarios del sistema. Cada archivo lleva como título el nombre de usuario de su propietario.
En el **Capítulo 4** aprendimos a enviar mensajes a los usuarios locales del sistema. En esta sección aprenderemos a enviar y recibir mensajes de correo electrónico a través de Internet.
El motivo de su ejecución constante como background en el sistema es que si surge la necesidad de enviar un e mail en cualquier momento, Sendmail estará allí cargado para realizar la petición. Si muchos usuarios quieren enviar e-mails, no se puede ejecutar Sendmail cada vez que se realiza la petición.
Sendmail es una gran aplicación (hay libros completos escritos sobre él), y su fama de dificultosa configuración tiene algo de cierta. Intentaremos hacerla lo más simple posible.
El archivo de configuración de Sendmail se encuentra en **/etc/sendmail.cf**. Si miran en su interior, puede que se asusten, es enorme. Los pasos por seguir para realizar una configuración básica de esta herramienta son:

Configurar Sendmail — PASO A PASO

1 Creen un usuario en el sistema con el mismo nombre de su cuenta de Internet, y que tenga acceso a los servicios de Internet.

2 Indiquen el servidor de correo para el envío de mensajes. Si no lo saben, consúltenlo a su ISP. Una vez obtenida la dirección, se busca la siguiente cadena dentro del archivo:

```
# "Smart" relay host (may be null)
DS
```

Y se le agrega la dirección del servidor pegada a la segunda línea. El resultado sería el siguiente:

```
# "Smart" relay host (may be null)
DSmail.suservidor.com
```

Si la segunda línea está comentada, quítenle el comentario.

3 Agreguen una definición para que en todos los mensajes que envíen se incluya la dirección de su ISP en la celda FROM. Para eso, busquen la línea:

```
# who I masquerade as (null for no masquerading) (see also $=M)
DM
```

Y modifíquenla por la siguiente:

```
# who I masquerade as (null for no masquerading) (see also $=M)
DMsuservidor.com
```

Siendo **suservidor.com** la dirección de su servidor que figura luego de la "@" en su dirección de e-mail.

4 Listo, ahora pueden enviar mails utilizando el comando **mail** seguido de la dirección de correo electrónico.

Si queremos enviar todos los mails que se encuentran en la cola de envío, tecleamos "**sendmail -q**". Si queremos consultarla, utilizamos el comando **mailq**.

Fetchmail

Esta aplicación se encuentra incluida en la gran mayoría de las distribuciones de GNU/Linux, como todas las comentadas en este libro. Su principal función es conectarse al servidor de e-mail y revisar si hay mensajes para el usuario en cuestión.
La configuración de este programa no es complicada. Los pasos por seguir son los siguientes:

Paso a paso — Configurar Fetchmail

1. Creen un archivo .fetchmailrc en su directorio **$HOME**.

2. En su interior, tipeen la siguiente linea:

```
poll mail.suservidor.com user usuario pass clave fetchall;
```

Siendo *"mail.suservidor.com"* la dirección de su servidor POP, *"usuario"* su nombre de usuario, y *"clave"* la palabra clave (password).

3. Otórguenle permisos exclusivos al usuario dueño (no más de **rw** para el dueño) de ese archivo, ya que en su interior contiene el nombre de usuario y la clave.

El archivo .fetchmailrc debe lucir como el siguiente:

```
# fetchmail control file sample (remove this header comment from yours!)
#
# This file (or one of your own creation, rather) should be located
# in your home directory with the name .fetchmailrc.  Permissions on this
# file may be no greater than -rw------- (0600), or fetchmail will refuse to
# use it.
#
# To see what effect your ~/.fetchmailrc file has, do
#
#    fetchmail -version
#
# This will display the fetchmail version number and an explanation
# in English of what the currently set options and defaults mean.
#
# Comments begin with a '#' and extend through the end of the line.
```

```
# Blank lines between server entries are ignored.
# Keywords and identifiers are case sensitive.
# When there is a conflict between the command-line arguments and the
# arguments in this file, the command-line arguments take precedence.
#
# Legal keywords are
#   poll                    - must be followed by a mailserver name or label
#   skip                    - must be followed by a mailserver name or label
#   via                     - must be followed by true name of host to poll
#   interval                - must be followed by an interval skip count
#   protocol (or proto)     - must be followed by a protocol ID
#   uidl
#   no uidl
#   port                    - must be followed by a TCP/IP port number
#   authenticate (or auth)  - must be followed by an authentication type
#   timeout                 - must be followed by a numeric timeout value
#   envelope                - must be followed by an envelope header name
#   qvirtual                - must be followed by a name prefix
#   no envelope
#   aka                     - must be followed by one or more server aliases
#   localdomains            - must be followed by one or more domain names
#   interface               - must be followed by device/IP address/mask
#   monitor                 - must be followed by device
#   plugin                  - must be followed by a string command name
#   plugout                 - must be followed by a string command name
#   dns
#   no dns
#   checkalias              - do multidrop address comparison by IP address
#   no checkalias           - do multidrop address comparison by name
#
#   username (or user)      - must be followed by a name
#   is                      - must be followed by one or more names
#   folder                  - must be followed by remote folder names
#   password (or pass)      - must be followed by a password string
#   smtphost (or smtp)      - must be followed by host names
#   smtpaddress             - must be followed by a host name
#   antispam                - must be followed by a numeric response value
#   mda                     - must be followed by an MDA command string
#   bsmtp                   - must be followed by a filename or -
#   lmtp
#   preconnect (or pre)     - must be followed by an executable command
```

```
#    postconnect (or post)    - must be followed by an executable command
#
#    keep
#    flush
#    fetchall
#    rewrite
#    forcecr
#    stripcr
#    pass8bits
#    dropstatus
#    mimedecode
#    no keep
#    no flush
#    no fetchall
#    no rewrite
#    no forcecr
#    no stripcr
#    no pass8bits
#    no dropstatus
#    no mimedecode
#    limit                    - must be followed by numeric size limit
#    warnings                 - must be followed by numeric size limit
#    fetchlimit               - must be followed by numeric msg fetch limit
#    batchlimit               - must be followed by numeric SMTP batch limit
#    expunge                  - must be followed by numeric delete count
#    properties               - must be followed by a string
#
# Legal protocol identifiers are
#    pop2 (or POP2)
#    pop3 (or POP3)
#    imap (or IMAP)
#    imap-k4 (or IMAP-K4)
#    apop (or APOP)
#    rpop (or RPOP)
#    kpop (or KPOP)
#    etrn (or ETRN)
#
#    Legal authentication types are
#    login
#    kerberos
#
```

```
#    set logfile             - must be followed by a string
#    set idfile              - must be followed by a string
#    set postmaster          - must be followed by a string
#    set daemon              - must be followed by a number
#    set syslog
#    set invisible
#
# The noise keywords `and', `with', `has', `wants', and `options' are ignored
# anywhere in an entry; they can be used to make it resemble English.  The
# punctuation characters `,' `:' `;' are also ignored.
#
# The run control file format is fully described (with more examples) on the
# fetchmail manual page.
#
# This is what the developer's .fetchmailrc looks like:

#set daemon 300     # Poll at 5-minute intervals

defaults
# Use this to test POP3
poll www.servidor.com.ar with protocol APOP;
     username usuario password el_password fetchall;
```

Una vez configurado, tipeen **fetchmail** para recibir los mensajes. Cuando el programa termine de descargar toda la mensajería, ejecuten el comando **mail** para poder leerlos.

El protocolo FTP

Otro de los servicios que Internet nos ofrece es el FTP (*File Transfer Protocol*). Este sistema nos permite realizar transferencias de archivos entre una máquina y otra. Generalmente, su estructura de funcionamiento se divide en dos partes: un servidor (*ftp server*), el cual soporta una cantidad limitada de usuarios y provee el acceso a los archivos; y un cliente (*ftp client*), que es el usuario que accede a ese servidor para enviar o recibir un archivo utilizando algún cliente FTP. Linux viene con un cliente llamado FTP, y es posible ejecutarlo tipeando desde la línea de comandos:

```
ftp direccion
```

Siendo **dirección** el IP o dominio del servidor FTP.
Una vez conectados al servidor, nos pedirá un nombre de usuario y su respectiva clave de acceso. En la mayoría de los servidores FTP, para obtener un acceso básico al servidor, el nombre de usuario es *"anonymous"* (usuario anónimo), y la clave, su dirección de e-mail.
Ya dentro de una sesión de FTP, el cliente tiene a su disposición una serie de comandos de navegación y transferencia para la ejecución de acciones sobre el servidor. Algunos de éstos son:

COMANDO	DESCRIPCIÓN
cd dir	Cambia al directorio "dir".
dir	Lista los archivos del directorio actual.
get archivo	Comienza la recepción del archivo "archivo".
put archivo	Comienza el envío del archivo "archivo".
ascii	Pasa a modo de transferencia ASCII.
binary	Pasa a modo de transferencia binaria.
delete	Borra un archivo remoto.
rename	Renombra un archivo remoto.
mkdir	Crea un directorio en el sistema remoto.
rmdir dir	Remueve el directorio "dir" en el sistema remoto.
beep	Emitir un beep cuando el comando haya terminado.
system	Muestra el tipo de sistema remoto.
size	Muestra el tamaño de un archivo remoto.
mget	Permite recibir múltiples archivos.
mput	Permite enviar múltiples archivos.
? comando	Muestra ayuda del comando "comando". Si no se le pasa ningún parámetro, muestra el listado de los comandos soportados.
quit	Cierra la sesión y sale del cliente.

Cabe destacar que algunos comandos (en su mayoría, de modificación de datos en el servidor), no están habilitados en los usuarios comunes.

Búsqueda de archivos: Archie

Existe un servicio en Internet llamado "Archie", que se encarga de buscar archivos en varios servidores FTP. Podemos utilizar este servicio para buscar un archivo, y nos devolverá un listado de sitios FTP en pocos segundos.
Un buen cliente es el que viene con el paquete de Internet de KDE; su interfaz es simple y rápida de utilizar. A continuación detallamos un listado de servidores Archie:

SERVIDOR	DIRECCIÓN IP	UBICACIÓN
archie.ac.il	132.65.20.254	(Israel)
archie.ans.net	147.225.1.10	(ANS, NY (USA))
archie.au	139.130.4.6	(Australia)
archie.doc.ic.ac.uk	146.169.11.3	(United Kingdom)
archie.edvz.uni-linz.ac.at	140.78.3.8	(Austria)
archie.funet.fi	128.214.6.102	(Finlandia)
archie.internic.net	198.49.45.10	(ATT, NY (USA))
archie.kr	128.134.1.1	(Corea)
archie.kuis.kyoto-u.ac.jp	130.54.20.1	(Japón)
archie.luth.se	130.240.18.4	(Swedish)
archie.ncu.edu.tw	140.115.19.24	(Taiwán)
archie.nz	130.195.9.4	(Nueva Zelanda)
archie.rediris.es	130.206.1.2	(Spain) (Éste es el nuestro)
archie.rutgers.edu	128.6.18.15	(Rutgers University (USA))
archie.sogang.ac.kr	163.239.1.11	(Korea)
archie.sura.net	128.167.254.195	(SURAnet (USA))
archie.sura.net(1526)	128.167.254.195	(SURAnet alt. MD (USA))
archie.switch.ch	130.59.1.40	(Swiss Server)
archie.th-darmstadt.de	130.83.22.60	(Alemania)
archie.unipi.it	131.114.21.10	(Italia)
archie.univie.ac.at	131.130.1.23	(Austria)
archie.unl.edu	129.93.1.14	(U. of Nebraska, Lincoln (USA))
archie.univ-rennes1.fr		(Francia)
archie.uqam.ca	132.208.250.10	(Canadá)
archie.wide.ad.jp	133.4.3.6	(Japón)

Este listado ha sido extraido del **FTP-Anonimo-COMO** del Infslug. Para obtener más información sobre este tipo de archivos, lean el último capítulo del libro.

En resumen...

En este capítulo hemos visto cómo usar las herramientas más importantes para trabajar con servicios del protocolo TCP/IP (comúnmente, herramientas de Internet). FTP, correo electrónico y Archie son algunos de los servicios que en general son más rápidos de usar en el entorno textual. Por esta razón, presentamos las aplicaciones más populares de Internet para la consola. En los próximos capítulos comenzaremos a montar nuestros propios servicios de red.

LA BIBLIA DE

Capítulo 8

Login remoto

Comenzaremos esta serie de capítulos dedicados a los entornos de redes hablando de un servicio sumamente útil: el login remoto.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

¿Qué es el login remoto?

Primero comenzaremos definiendo la palabra *login* como el proceso de registración de un usuario en el sistema. El momento de login (o "logueo") es cuando el sistema pide un nombre de usuario y una contraseña al operador, para poder usarlo (recordemos que GNU/Linux es un sistema multiusuario).
Cada usuario del sistema posee un perfil diferente, que está compuesto por:

- Su propia configuración del entorno de trabajo.
- Sus propias herramientas.
- Sus propios límites de acceso a datos.

Ahora bien, en cualquier sistema, un solo usuario puede estar trabajando frente al teclado, pero por medio de la red, muchos usuarios pueden estar conectados a ese sistema y utilizarlo tal como si tuvieran las puntas de sus dedos sobre el teclado del servidor. Una vez que el usuario se registra por medio de la red en el servidor, ya puede utilizar el sistema con su propia configuración del entorno de trabajo, sus propias herramientas y sus límites de uso. Al mismo tiempo, otros usuarios pueden estar trabajando simultáneamente, y se crea algo así como un centro de concentración de usuarios donde, además de trabajar y usar sus herramientas, pueden establecer conversaciones y enviarse mensajes.
Existen, básicamente, dos protocolos que nos permiten realizar esta tarea: el primero es el legendario Telnet, y el segundo, SSH (*Secure Shell*). Analizaremos el uso de ambos.

Telnet

Éste es uno de los servicios más antiguos y utilizados del protocolo TCP/IP. Telnet funciona abriendo el puerto 23 del sistema y esperando peticiones de registro por su parte. Para que el sistema acepte este tipo de conexiones, deberá estar ejecutando el demonio de Telnet (**telnetd**). Si poseen **Inetd**, pueden habilitarlo en el archivo **/etc/inetd.conf**, o pueden también hacer uso del gestor de servicios de su distribución favorita. Básicamente, para utilizar este servicio, deberán instalar los paquetes **telnet-server** y **telnet-client**.
El protocolo Telnet es muy fácil de instalar (sólo es necesario descomprimir los paquetes), y no posee un sistema de configuración. Por esa razón, todo parece ser totalmente automático.
Una vez que el servidor está corriendo, los usuarios ya pueden comunicarse con él mediante un *cliente de telnet*, el cual es totalmente independiente del sistema

operativo que esté utilizando. Esto significa que se puede acceder a un sistema GNU/Linux con un cliente de Telnet para Mac OS X, por ejemplo. En GNU/Linux, el cliente de Telnet se utiliza del siguiente modo:

```
telnet [host/ip]
```

Una vez establecida la comunicación, el sistema mostrará el mensaje de bienvenida (definido por el administrador del servidor en el archivo **/etc/issue.net**) y preguntará por un nombre de usuario y una contraseña. Si todo sale correctamente, el usuario ya podrá comenzar a hacer uso del sistema mediante los diferentes comandos que tiene a su disposición, la mayoría de los cuales hemos analizado en este libro.
Para salir de Telnet, el usuario sólo tiene que cerrar el shell del sistema, con el comando **exit**.
Lo interesante de este servicio es que, al formar parte del protocolo TCP/IP, puede ser utilizado a través de Internet. O sea, que una persona en Nueva York podría estar utilizando su sistema instalado en Tokio de forma totalmente transparente y con las limitaciones de velocidad que ofrezca el tipo de conexión de acceso a la red.

Secure Shell

El sistema SSH es similar en funcionamiento a Telnet: permite el login remoto a un sistema. Pero la diferencia principal es que SSH es un sistema sumamente seguro, ya que admite diferentes algoritmos de encriptación, como BlowFish, DES, IDEA y RSA.

Tal como dice el RFC de Secure Shell: *"SSH (Secure Shell) es un programa para conectarse remotamente a otra computadora a través de una red, para ejecutar comandos en esa máquina remota y para transferir archivos de una máquina a otra. Proporciona una exhaustiva autenticación y comunicaciones seguras en redes no seguras"*.

SÓLO PARA EL MODO TEXTO

Cabe destacar que el servicio de login remoto es sólo funcional en el modo texto. Los usuarios podrán trabajan con herramientas de consola y no con herramientas del modo gráfico (Xfree86). Para esto tendrán que usar otro sistema como, por ejemplo, el VNC.

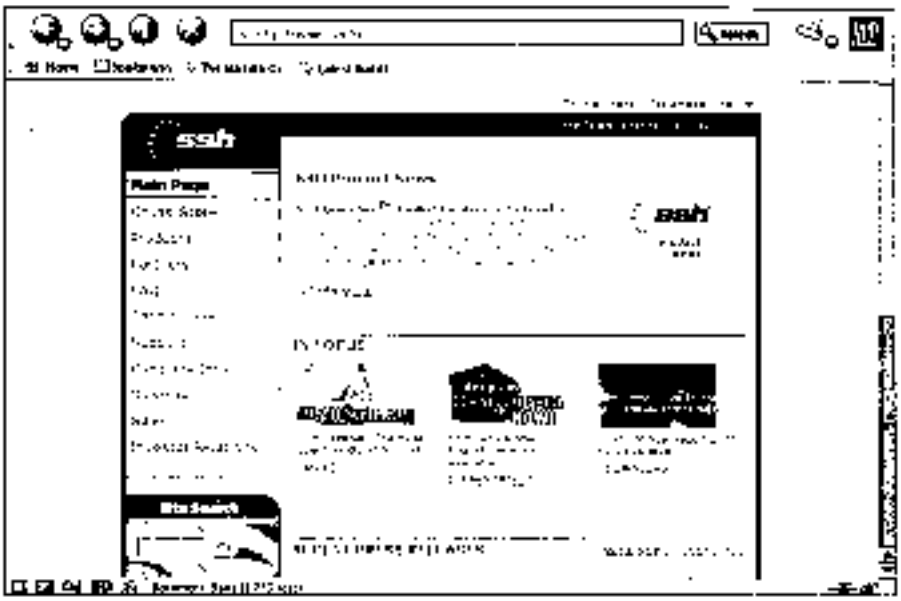


Figura 1. En el sitio ***www.ssh.fi*** *encontrarán mucha información sobre este sistema de login remoto seguro.*

Todas estas características hacen que a la hora de utilizar SSH, éste parezca bastante más lento que Telnet. Esto es porque todos los datos que está transfiriendo por la red están siendo encriptados de un lado y desencriptados del otro. Obviamente, toma tiempo, pero aumenta muchísimo la seguridad de la privacidad de sus datos.

El demonio sshd

El servidor de Secure Shell se llama **sshd**, y se encuentra incluido en casi todas las distribuciones actuales de GNU/Linux. Pero si lo quieren instalar por su cuenta, pueden hacerlo descargando la última versión de **www.ssh.fi** y compilando el código fuente (no tienen más que hacer **./configure**, **make** y **make install**).
Una vez que lo tengan instalado, pueden ejecutarlo directamente (comando **sshd**), o pueden verificar si ya está funcionando en sus sistemas con el comando:

```
ps ax | grep sshd
```

Si todo está en orden, ya pueden proceder a usar el servicio.

El cliente de SSH

Nuevamente, el cliente es una aplicación totalmente independiente del sistema operativo. En GNU/Linux, el cliente se llama, simplemente, **ssh**.

Para comunicarse con un servidor SSH, simplemente deben utilizar el siguiente comando:

```
ssh usuario@host
```

o

```
ssh -l usuario [host/IP]
```

Ahora sí, ya podrán comenzar a trabajar con este sistema de login remoto de forma mucho más segura que en Telnet.

Comunicación entre usuarios

Los usuarios que estén trabajando en el sistema (no importa por medio de qué protocolo sea) tienen a su disposición una serie de herramientas para la comunicación entre ellos. Veamos cuáles son.

• Si los usuarios no están en el sistema en el mismo momento, pueden enviarse un mail (correo electrónico) utilizando el comando **mail** del siguiente modo:

```
mail username
```

Por ejemplo: si el usuario *root* quiere enviar un mail a *trax*, sólo deberá tipear:

```
mail trax
Subject: Trabajo
Simplemente quería recordarte que necesito el trackshoot de la ultima
mezcla para esta tarde.
[CTRL-D]
Cc: [ENTER]

[root@localhost /]#
```

Como vemos, el comando **mail** nos permite enviarle un mail a un usuario. El campo *subject* es el título del mensaje. Luego se escribe el cuerpo del mensaje (o sea,

el mensaje en sí). Para finalizar la escritura del mensaje, se tipea la combinación de teclas **CTRL+D**.
A continuación, **mail** presenta otro campo (**Cc:**), en el cual deberemos ingresar el nombre de otro usuario (para enviarle una copia del mensaje), o presionar **ENTER**, en caso de no necesitarlo.
Concluido el proceso, cuando el usuario *trax* se registre, el sistema le mostrará el siguiente mensaje:

```
You have new mail.
```

Para leer el mensaje, trax sólo tendrá que utilizar el comando **mail**. Éste, como resultado, mostrará una lista de mensajes similar a la siguiente:

```
Mail version 8.1 6/6/93. Type ? for help
"/var/spool/mail/trax": 1 message new
>N 1 root@localhost Thu Jan 20 10:23  13/427  "Trabajo"
&
```

En este listado podemos ver de quién vienen los mensajes, y fecha, hora y título de éstos. Para leerlos, presionamos el valor correspondiente a cada mensaje. En el ejemplo, el número de mensaje es el 1.
Una vez leído, podemos presionar el número de otro mensaje, leer la lista de mensajes nuevamente (presionando la letra "h") o salir del programa presionando la tecla "q".

Envío de mensajes instantáneos

Si los usuarios están operando en el sistema al mismo tiempo (en diferentes terminales), pueden enviarse mensajes utilizando el comando **write**. El formato es el siguiente:

```
write usuario
Mensaje (...)
[CTRL-D]
```

Es posible que cuando queramos enviar un mensaje, el sistema nos devuelva el siguiente error:

```
write: you have write permission turned off.
```

Esto se debe a que no está habilitado el permiso para escribir mensajes. Para habilitarlo, utilizamos el comando **mesg**, el cual posee el siguiente formato:

```
mesg [y/n]
```

Para habilitar el envío y la recepción de mensajes instantáneos, simplemente hay que tipear:

```
mesg y
```

Cabe destacar que ambos usuarios deben tener este permiso habilitado para poder establecer una comunicación entre ellos.

Comunicaciones de tipo chat

Linux provee una utilidad para *chatear* entre los usuarios del sistema denominada talk. Para establecer una comunicación, uno de los dos usuarios debe comenzar tipeando una linea en el siguiente formato:

talk usuario@localhost

Y esperará a que el otro usuario responda tipeando el mismo comando (obviamente, cambiando el nombre de usuario). Una vez establecida la conexión, los dos usuarios pueden intercambiar mensajes en tiempo real.
Para terminar la sesión, pueden oprimir la combinación de teclas **CTRL+C** y de esa forma matar ese proceso.

Enviando mensajes colectivos

Los mensajes colectivos son mensajes que el administrador envia a todos los usuarios en un momento dado. Esto es útil para avisar cuando se va a cerrar el sistema, para notificar alguna modificación sobre los archivos o, simplemente, mencionar algo.
Para realizar esta operación, existe un comando de fácil utilización llamado **wall** (*write all*). El formato de este comando es el siguiente:

```
wall mensaje
```

De esta manera, si se quiere enviar un mensaje diciendo "Cerraré el sistema en cinco minutos", sólo se deberá ejecutar:

```
wall Cerrare el sistema en 5 minutos. El administrador
```

Y a todos los usuarios conectados en ese momento les saldrá un mensaje en pantalla:

```
Broadcast message from root: Cerrare el sistema en 5 minutos.
```

Pero es importante recordar que no se debe abusar de este comando: puede resultar molesto para sus usuarios estar recibiendo mensajes que posiblemente no tengan mucha importancia en períodos cortos de tiempo.

En resumen...

El acceso remoto a travees protocolos como Telnet o SSH permite controlar una terminal de GNU/Linux a distancia, tal como si tuvieran sus manos sobre el teclado del servidor. Esto es sumamente útil para realizar administraciones, revisión de errores y generación de reportes en horarios en los que no se está en el lugar de trabajo, por ejemplo. Como conclusión final, recomendamos para esta aplicación usar el protocolo SSH y no Telnet, ya que éste no posee encriptación de datos y, como consecuencia, expone más la comunicación ante intrusos que accedan a la red.

LA CONFIGURACIÓN DE SSHD

El demonio sshd posee un archivo de configuración ubicado en /etc/ssh_config. Desde allí, ustedes podrán definir, entre otras cosas, el sistema de encriptación que se usará, qué hosts pueden acceder al servidor, cuáles no, etc.

LA BIBLIA DE

Capítulo 9

Instalar un servidor web

Apache es uno de los servidores web más populares para Internet. En este capítulo, aprenderemos a instalarlo y a configurarlo.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@redusers.com

Descarga e instalación de Apache

Una de las aplicaciones más populares del sistema GNU/Linux es su utilización como base para un servidor web. ¿Por qué? Sencillo: porque estamos hablando de un sistema libre, muy robusto y totalmente preparado para manejarse en entornos de red donde la cantidad de clientes es, al menos, impredecible.

El servidor web más popular en el ámbito de los UNIX es Apache (a estas alturas, ¿quién no ha oído hablar de él?). Apache es un proyecto Open Source (posee su propia licencia de uso y distribución, para nada restrictiva) que hace ya varios años que está marchando por el camino correcto. Actualmente se están desarrollando dos series de versiones: la 2.0 y la 1.3. Nosotros nos dedicaremos a la instalación y el uso de la 1.3.

Vamos a comenzar descargando la última versión estable de la serie 1.3. Los pasos por seguir son generalmente los mismos en ambas versiones.

Para descargarlo, diríjanse al sitio **www.apache.org/dist** y bajen la última versión de la serie 1.3. El archivo tendrá una estructura de tipo **apache_1.3.XX.tar.gz**.

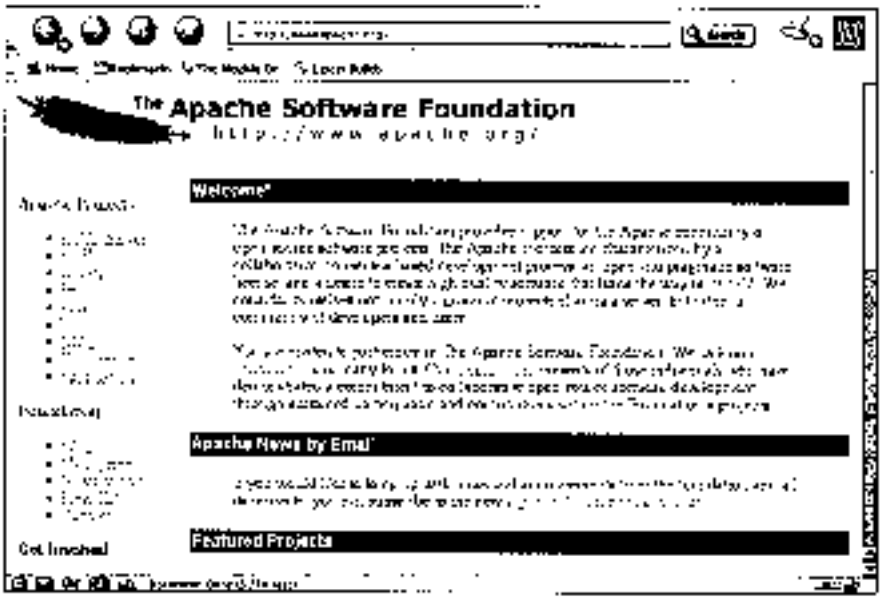


Figura 1. En el sitio oficial del proyecto Apache (www.apache.org) podrán encontrar la licencia bajo la que está regida esta aplicación.

Ahora copiaremos ese paquete a un directorio temporal y lo descomprimiremos. El archivo que acabamos de descargar es código fuente, y es preciso que lo compilemos.

```
mkdir /temporal
cp apache_1.3.XX.tar.gz /temporal
cd /temporal
tar -xzvf apache_1.3.XX.tar.gz
```

Con estos comandos ya tendremos todo el código fuente de Apache listo para ser compilado. El próximo paso consistirá en compilarlo.
Procederemos a configurarlo y compilarlo del siguiente modo (sitúense en el subdirectorio creado por **tar**, dentro del directorio temporal que creamos).

```
./configure -prefix="/usr/local/apache"
make
make install
```

El proceso no es en absoluto complicado, y tardará algunos minutos en terminar. Una vez que haya concluido, ustedes ya tendrán su servidor web instalado en **/usr/local/apache**. ¡Felicitaciones!

Configuración del servidor

Apache almacena toda la configuración relativa a su funcionamiento en el archivo localizado en **/usr/local/apache/conf/httpd.conf**. Este archivo es un gran archivo de texto con muchas opciones para personalizar. Detallaremos sus partes más importantes. El archivo comienza con un texto introductorio:

```
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://www.apache.org/docs/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# After this file is processed, the server will look for and process
# /usr/local/httpd/conf/srm.conf and then /usr/local/httpd/conf/access.conf
# unless you have overridden these with ResourceConfig and/or
# AccessConfig directives here.
#
# The configuration directives are grouped into three basic sections:
```

```
#  1. Directives that control the operation of the Apache server process as a
#     whole (the 'global environment').
#  2. Directives that define the parameters of the 'main' or 'default' server,
#     which responds to requests that aren't handled by a virtual host.
#     These directives also provide default values for the settings
#     of all virtual hosts.
#  3. Settings for virtual hosts, which allow Web requests to be sent to
#     different IP addresses or hostnames and have them handled by the
#     same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/usr/local/apache" will be interpreted by the
# server as "/usr/local/apache/logs/foo.log".
#
```

Luego, ingresa en la primera sección, llamada "entorno global". Aquí definiremos, entre otras cosas, cómo correrá el servidor, en dónde están los archivos del servidor, etc.

```
### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#
#
# ServerType is either inetd, or standalone.  Inetd mode is only supported on
# Unix platforms.
#
ServerType standalone
```

Tal como lo dice el comentario, en la directiva **ServerType** podemos configurar cómo funcionará el servidor Apache. Existen dos formas posibles: **standalone** e **inetd**. Si elegimos la primera de ellas, Apache funcionará como un proceso independiente y deberemos encenderlo y apagarlo utilizando sus comandos especiales. Si usamos la

segunda opción, entonces Apache estará preparado para ser gestionado por **inetd**, un gestor de servicios de red. La configuración de **inetd** se puede encontrar en el archivo **/etc/inetd.conf**. Es recomendable dejar esta opción tal como está.

```
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
ServerRoot "/usr/local/httpd"
```

En la directiva **ServerRoot** definimos el directorio principal en el cual está instalado Apache. Recordemos que, al configurarlo, definimos que deseábamos instalarlo en **/usr/local/apache**. Es posible que sea necesario modificar esta opción para que se adecue a nuestras circunstancias.

```
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
PidFile /var/run/httpd.pid
```

En la directiva **PidFile** podemos definir en qué archivo Apache almacenará su número de identificación de proceso (PID) cuando se encienda. Es recomendable dejar este valor intacto.

```
#
# ScoreBoardFile: File used to store internal server process information.
# Not all architectures require this.  But if yours does (you'll know because
# this file will be  created when you run Apache) then you *must* ensure that
# no two invocations of Apache share the same scoreboard file.
#
ScoreBoardFile /var/run/httpd.scoreboard
```

Otro archivo de definiciones internas. Éste almacena información interna de procesos. Sigamos dejando todo como está si no queremos tener problemas.

```
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300
```

Aquí podemos definir el número de segundos que el servidor Apache esperará antes de recibir o enviar un mensaje de **Timeout** (tiempo agotado).

```
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
```

Esto permite activar o no la aceptación de conexiones persistentes. Éstas son las conexiones que envían varios mensajes de petición (varios request). Si no queremos aceptar este tipo de conexiones, simplemente tendremos que tipear **Off** en su valor de configuración.

```
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100
```

Y si aceptamos las conexiones persistentes, debemos definir cuántas peticiones admitiremos como máximo por conexión. Quizá 100 sea un número demasiado grande, aunque esto variará según sus necesidades y sus recursos. Utilicen el valor 0 para permitir peticiones ilimitadas (mucho cuidado con esto).

```
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15
```

Aquí podemos definir el número de segundos que se esperará para la próxima petición del mismo cliente en la misma conexión. Si tienen una conexión lenta a Internet, quizá convenga elevarlo un poco.

```
#
# Server-pool size regulation.  Rather than making you guess how many
# server processes you need, Apache dynamically adapts to the load it
# sees -- that is, it tries to maintain enough server processes to
# handle the current load, plus a few spare servers to handle transient
# load spikes (e.g., multiple simultaneous requests from a single
# Netscape browser).
#
# It does this by periodically checking how many servers are waiting
# for a request.  If there are fewer than MinSpareServers, it creates
# a new spare.  If there are more than MaxSpareServers, some of the
# spares die off.  The default values are probably OK for most sites.
#
MinSpareServers 1
MaxSpareServers 1
```

Esta opción es bastante interesante. Se encarga automáticamente de encender y apagar una serie de servidores "de repuesto". De esta forma, el servicio web está siempre preparado para varios clientes, no importa cuántos sean. Si el número de servidores de repuesto es menor que el definido en **MinSpareServers**, entonces Apache creará un nuevo servidor. Ahora, si hay más servidores que los definidos en **MaxSpareServers**, borrará algunos. Como está configurado en este ejemplo, el sistema está siempre listo con un servidor de repuesto.

```
#
# Number of servers to start initially -- should be a reasonable ballpark
# figure.
#
StartServers 1
```

Aquí definimos el número de servidores que serán iniciados la primera vez que se ejecute Apache. Es recomendable que eleven este número a 7 u 8, o a un valor mayor si creen que pueden llegar a tener más conexiones simultáneas en el inicio del servicio.

```
#
# Limit on total number of servers running, i.e., limit on the number
# of clients who can simultaneously connect -- if this limit is ever
# reached, clients will be LOCKED OUT, so it should NOT BE SET TOO LOW.
# It is intended mainly as a brake to keep a runaway server from taking
# the system with it as it spirals down...
#
MaxClients 150
```

Aquí definimos el número máximo de clientes que se pueden comunicar con nuestro servicio web. Si el límite es excedido, los clientes no podrán acceder al servicio. Por eso es recomendable que este valor sea elevado para no tener problemas, aunque nuevamente, esto dependerá de los recursos del sistema. Esto es, si tienen una máquina potente, no duden en elevarlo.

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
Listen 80
```

Aquí definimos en qué puerto Apache escuchará las peticiones. El puerto 80 es el estándar en el cual se encuentran todos los servidores web de Internet, aunque para una intranet, quizá quieran cambiar el valor por razones de seguridad. De todas formas, si alteran este valor, los clientes tendrán que definirlo en la dirección URL de sus navegadores de la siguiente forma:

[direccionIP/host]:[numero_de_puerto]

```
#
# BindAddress: You can support virtual hosts with this option. This directive
# is used to tell the server which IP address to listen to. It can either
# contain "*", an IP address, or a fully qualified Internet domain name.
# See also the <VirtualHost> and Listen directives.
#
#BindAddress *
```

En esta opción podemos definir que Apache escuche peticiones desde una dirección IP determinada o desde un nombre de dominio de Internet. Esta opción está comentada para que todo el mundo pueda tener acceso a nuestro servidor web.

A continuación se definen todas las librerías dinámicas que serán usadas en Apache. Recortaremos el listado porque es muy largo.

```
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Please read the file README.DSO in the Apache 1.3 distribution for more
# details about the DSO mechanism and run `httpd -l' for the list of already
# built-in (statically linked and thus always available) modules in your httpd
# binary.
#
# Note: The order in which modules are loaded is important.  Don't change
# the order below without expert advice.
#
# Example:
# LoadModule foo_module libexec/mod_foo.so
LoadModule mmap_static_module /usr/lib/apache/mod_mmap_static.so
LoadModule vhost_alias_module /usr/lib/apache/mod_vhost_alias.so
LoadModule env_module         /usr/lib/apache/mod_env.so
LoadModule define_module      /usr/lib/apache/mod_define.so
LoadModule config_log_module  /usr/lib/apache/mod_log_config.so

(...)

#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
#
ExtendedStatus On
```

Aquí podemos definir si Apache genera reportes extensos sobre el estado de su funcionamiento o, simplemente, muestra la información básica, cuando el administrador "server-status" es llamado.

Lo más importante

Y ahora sí, hemos llegado a la configuración principal del servidor. Le diremos a Apache a dónde tiene que ir a buscar los archivos HTML, cuál es la dirección de mail del administrador, en dónde se almacenarán los archivos de registro, y mucho, pero mucho más. Comencemos.

```
### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
# Port: The port to which the standalone server listens. For
# ports < 1023, you will need httpd to be run as root initially.
#
Port 80
```

Aquí definimos en qué puertos Apache escuchará las peticiones cuando funciona en modo **standalone**. Si lo pusieron bajo **inetd**, esta directiva no tendrá efecto, ya que el tema de los puertos es gestionado por el mismo demonio **inetd**.

```
##
##  SSL Support
##
##  When we also provide SSL we have to listen to the
##  standard HTTP port (see above) and to the HTTPS port
##
<IfDefine SSL>
Listen 80
Listen 443
</IfDefine>
```

Aquí definimos por qué puertos aceptaremos SSL (conexiones seguras). Los dos definidos en el ejemplo son los estándar.

```
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
#  . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#  . On HPUX you may not be able to use shared memory as nobody, and the
#    suggested workaround is to create a user www and use that user.
#  NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group nogroup on these systems!
#
User wwwrun
Group nogroup
```

En esta directiva podemos definir con qué usuario y bajo qué grupo correrá el servidor Apache. Obviamente, nunca es recomendable que el servidor corra como usuario root.

```
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents.
#
ServerAdmin root@localhost
```

En esta directiva se define cuál es la dirección de correo electrónico del administrador del servidor. A esta dirección llegarán los mensajes de error, y es la dirección que se mostrará a los clientes cuando algo funcione mal.

¿POR QUÉ DOS VERSIONES?

Mientras la gente sigue usando la versión 1.3, los desarrolladores de Apache publicaron la versión 2.0. Esta versión, con muchas mejoras, no fue evaluada en su totalidad, y la gente aún no posee demasiada confianza en él. Por esta razón, los miembros del proyecto Apache decidieron seguir con el desarrollo de la serie 1.3 para seguir soportando a toda esta cantidad de usuarios que ya lo utilizan, mientras esperamos a que la versión 2 sea evaluada y desarrollada con un poco más de profundidad.

```
#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
# 127.0.0.1 is the TCP/IP local loop-back address, often named localhost. Your
# machine always knows itself by this address. If you use Apache strictly for
# local testing and development, you may use 127.0.0.1 as the server name.
#
ServerName miservidor
```

En **ServerName** definimos cuál es el nombre de host de nuestro servidor. Si tenemos un dominio de Internet registrado, puede ser ese mismo. Si no, podemos usar nuestra dirección IP o el nombre de host que definimos en GNU/Linux.

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/usr/local/apache/htdocs"
```

En la directiva **DocumentRoot** podemos definir en qué directorio serán almacenadas las páginas web. Si quieren guardar las páginas en otro directorio, entonces tienen que modificar esta directiva.
El directorio **/usr/local/apache/htdocs** será donde deberán almacenar el **index.html**, que se verá en pantalla cuando los clientes tipeen su dirección de IP o su nombre de host en sus navegadores.

```
#
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
#
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>
```

Si queremos que los usuarios de nuestro servidor GNU/Linux puedan tener sus propias páginas web, entonces debemos definir en la directiva **UserDir** cuál es el directorio en el que ellos deben almacenar sus páginas. En el caso del ejemplo, cada usuario deberá crear un directorio llamado **public_html**, en donde guardará sus páginas web. Para acceder a estas páginas, los clientes deberán usar la siguiente estructura en la dirección URL de sus navegadores web:

```
http://[direccion]/~[usuario]
```

Es necesario que los usuarios otorguen permiso de lectura y ejecución a este directorio, para que los visitantes puedan leer su contenido.

```
#
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index.  Separate multiple entries with spaces.
#
<IfModule mod_dir.c>
    DirectoryIndex index.html
</IfModule>
```

Aquí se define cuál es el archivo índice. No creo que a nadie se le ocurra cambiar esta directiva.

UNA PECULIAR FORMA DE INSTALACIÓN

Vamos a configurar Apache para que se instale todo en un solo directorio, en lugar de distribuirse por todo el árbol de directorios de nuestro sistema. De esta forma, ustedes siempre tendrán los archivos a mano (y en un mismo lugar), y el día que decidan desinstalarlo, simplemente tendrán que eliminar ese directorio.

```
#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off
```

Cuando se almacena la información de registro de los clientes que acceden a nuestro servidor, podemos indicarle a Apache que guarde su dirección IP o su nombre de host. Esta opción está deshabilitada por defecto, ya que es lo mejor para la red en general.

```
#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog /var/log/httpd/error_log
```

En esta sección definimos en qué archivo serán almacenados los mensajes de error.

```
#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn
```

También podemos definir con cuántos detalles serán almacenados los mensajes de registro. Hay varias posibilidades: **debug**, **info**, **notice**, **warn**, **error**, **crit**, **alert** y **emerg**. Los nombres son bastante descriptivos, y es sólo cuestión de definir el que más se adecue a nuestras necesidades.

```
#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
CustomLog /var/log/httpd/access_log common
```

En este archivo se almacenarán los registros de los clientes que acceden a nuestro servidor web. Aquí encontraremos información como la de sus respectivas direcciones IP, el día y la hora en que realizaron la petición, y qué tipo de petición enviaron.

```
#
# Customizable error response (Apache style)
#  these come in three flavors
#
#    1) plain text
#ErrorDocument 500 "The server made a boo boo.
#  n.b.  the single leading (") marks it as text, it does not get output
#
#    2) local redirects
#ErrorDocument 404 /missing.html
#  to redirect to local URL /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
#  N.B.: You can redirect to a script or a document using server-side-
includes.
#
#    3) external redirects
#ErrorDocument 402 http://some.other_server.com/subscription_info.html
#  N.B.: Many of the environment variables associated with the original
#  request will *not* be available to such a script.
```

En esta dirección podemos definir diferentes formas de mostrar un mensaje de error al cliente una vez que la falla se produce. Será cuestión de descomentar la que más nos agrade. Es posible definir un archivo al cual será reenviado el cliente si las cosas no funcionaron bien. Por ejemplo:

```
ErrorDocument 404 /perdido.html
```

Si tenemos un error 404 (página no encontrada), el cliente será reenviado a la página **perdido.html**.
A continuación, viene la sección de definición de hosts virtuales.

```
### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at <URL:http://www.apache.org/docs/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# Use name-based virtual hosting.
#
#NameVirtualHost *

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost dummy-host>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

#<VirtualHost _default_:dummy-host>
#</VirtualHost>
```

Finalmente, en la sección de hosts virtuales podemos definir varios servidores virtuales. Esto es, Apache puede responder a varias peticiones dirigidas a diferentes hosts. Para cada host, hay que definir las opciones esenciales, como dirección de e-mail del administrador, directorio donde están los archivos HTML, el nombre de host y los archivos en los que se almacenarán los archivos de registro.
Como habrán visto, este archivo es muy largo (en realidad, es más largo de lo que pudimos listar en este libro), y hay muchas opciones para personalizarlo al máximo. En realidad, con sólo modificar las opciones **ServerAdmin** y **ServerName**, ya tendrán Apache funcionando correctamente, aunque si quieren que su funcionamiento sea un poco más adecuado a sus necesidades, habrá que modificar algunas de las opciones que hemos listado en este libro.
Ahora que está todo configurado, el próximo paso es encender el servidor.

Encendido y apagado del servidor

Una vez que ya tienen su archivo **httpd.conf** configurado como más les agrade, ya pueden proceder a encender el servidor. Es decir, ejecutarlo.
En **/usr/local/apache/bin** están los binarios de nuestro servidor web. Allí encontraremos un pequeño script llamado **apachectl**. Éste posee, básicamente, tres parámetros: **start**, **restart** y **stop**. El primero de ellos enciende el servidor httpd, el segundo lo reinicia y el tercero lo para (apagado del servidor). Entonces, para encender nuestro flamante servidor web, tendremos que ejecutar:

```
apachectl start
```

Ahora ya pueden comenzar a armar sus páginas web y albergarlas en **/usr/local/apache/htdocs**. Mientras, pueden probar que todo funcione correctamente accediendo a su propia dirección IP (o a **127.0.0.1**) desde cualquier navegador que tengan instalado.

CHAU, APACHE

Recuerden que cuando quieran desinstalar su servidor Apache, sólo tendrán que eliminar el directorio /usr/local/apache. Esto eliminará tanto los archivos del servidor como las páginas web que haya contenido.

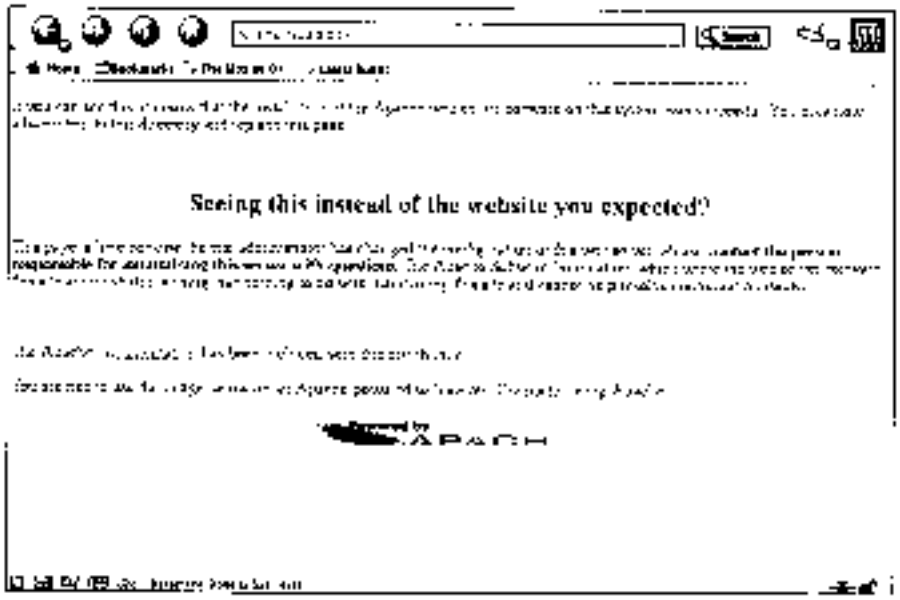


Figura 2. Una vez instalado y ejecutado el servidor web, una página temporal será instalada en nuestro sistema.

En resumen...

Apache es uno de los servidores más populares de Internet. Más del 60% de los web-servers que ocupan la Red funciona bajo este sistema, sobre GNU/Linux o cualquiera de los otros sistemas operativos soportados. Lo más interesante de todo es que los conocimientos vertidos en este capítulo les servirán para aplicarlos en cualquier otro sistema operativo soportado por Apache, ya que en todos ellos, el archivo de confrgu ración **httpd.conf** es igual. En los próximos capitulos seguiremos analizando otros servicios de red que, en complemento con los ya vistos, les permitirán tener un servidor completo y sumamente funcional.

LA BIBLIA DE

Capítulo 10

El servicio NFS

Mediante el servicio de compartimiento de archivos NFS (Network File System), podremos transferir archivos a través de una red, de forma rápida y transparente.

SERVICIO DE ATENCIÓN AL LECTOR lectores@tectimes.com

¿Qué es NFS?

El sistema de archivos de red NFS (Network File System, o sistema de archivos de red) nos permite compartir archivos y directorios entre computadoras de una forma rápida de instalar, fácil de administrar y de usar. Si bien es un sistema nativo de UNIX, existen clientes de NFS para Windows, por lo que también podrán transferir archivos entre los dos sistemas operativos con este sistema (aunque en este caso recomiento usar, obviamente, Samba).
Se trata de un sistema que trabaja bajo el modelo cliente/servidor. Esto es, una máquina actúa como servidor de archivos, y las demás máquinas que quieren acceder a ese espacio de trabajo compartido se llaman clientes. En este capitulo vamos a aprender a instalar un servidor basado en este servicio, asi como las formas de acceder a los recursos exportados desde el lugar de los clientes de la red.
NFS es un protocolo RPC que corre sobre el UDP. Un RPC es algo asi como una llamada al sistema que se ejecuta de forma remota.
Por otro lado, los demonios que trabajan con las peticiones de los clientes NFS no tienen asignados números de puertos UDP comunes y corrientes, sino que son asignados dinámicamente; por eso, comúnmente se denominan *RPC portmapper* (si buscan, encontrarán el proceso **portmap** corriendo en su sistema mientras NFS está activado).
Si quieren ver un listado de los puertos asignados por **portmap** a los diferentes servicios, no tienen más que ejecutar el comando **rpcinfo -p**.

Instalación de NFS

En general, el sistema NFS se instala de forma automática en casi todas las distribuciones de GNU/Linux. Pero generalmente no se activa por defecto en el inicio, ya que no todos los usuarios lo utilizan y, como todo proceso, consume recursos del sistema. Para saber si tienen el sistema de archivos NFS corriendo actualmente, lo único que tienen que hacer es pedir un listado de procesos con los siguientes parámetros:

```
ps ax | grep nfs
```

En caso de que el servicio NFS esté activado y corriendo, verán una serie de procesos (generalmente son ocho) llamados **[nfsd]**.
Si no aparece nada, entonces deberian verificar que el sistema NFS esté instalado; pueden hacerlo buscando la cadena "nfs" en el administrador de paquetes de la distribución que usen. En caso de no tener los paquetes instalados, inserten el CD de su

distribución e instalen todos los paquetes del sistema NFS. Pero si está instalado, entonces deberán activarlo para que se inicie automáticamente en el nivel de ejecución que usen. Las distribuciones basadas en Red Hat (como Mandrake, Conectiva y Turbo-Linux) poseen una herramienta llamada **setup**, que cuenta con una sección de configuración de servicios del sistema, en donde podemos activar la opción **NFS**. En SuSE Linux pueden activarlo con **yast**.
Existe una forma de encender el servidor NFS de forma manual. Ésta consiste en teclear el siguiente comando.

```
/etc/rc.d/init.d/nfs start
```

Obviamente, si usan este comando, tendrán que teclearlo cada vez que reinicien la máquina y quieran usar el sistema de archivos NFS. La forma "sucia" (pero completamente funcional, también) es agregar ese comando al final del archivo **/etc/rc.d/rc.local**.

Exportando directorios

En la jerga de NFS, exportar significa configurar un directorio para que los clientes de la red puedan tener acceso a él. Sería algo así como compartir un directorio (algunas veces, hay que usar términos vulgares para explicar mejor la idea de algún concepto no muy complicado de entender).
Para realizar esta acción, debemos configurar NFS de forma adecuada. Todo se realiza desde un archivo localizado en **/etc/exports**, el cual generalmente está vacío. La estructura de sintaxis de este archivo es muy sencilla:

```
directorio_a_exportar host1(opciones)
```


NO REINICIEN NFS

No hace falta que reinicien el servidor NFS luego de hacer las modificaciones necesarios al archivo exports. Con actualizar la base de datos (comando exportfs) será más que suficiente. De todas formas, muchas distribuciones ejecutan el comando exportfs de manera automática al inicio.

El **directorio_a_exportar** es la cadena completa al directorio que queremos exportar. Es una mala idea exportar directorios importantes del sistema como **/bin** o **/root**. Lo mejor en estos casos es crear un directorio específico para NFS en **/home** y poner allí todos los archivos que se quieren servir.
Luego, hay que definir quiénes podrán acceder a este directorio (variable **hosts**). Aquí, el sistema es muy flexible, y podemos usar nombres de red, direcciones de IP y asteriscos para designar múltiples sistemas.
Por último, entre paréntesis hay que poner las opciones de compartimiento. Éstas generalmente son **ro** (compartir un directorio en modo de sólo lectura) y **rw** (compartir un directorio en modo lectura/escritura). Hay más opciones, que podrán encontrar en las páginas del manual de este archivo (**man exports**).
Algunos ejemplos de entradas en el archivo **/etc/exports** son los siguientes:

```
/home/paracompartir        *(ro)
/home/paracompartir2       10.0.0.*(rw)
/home/paracompartir3       maquina1(ro) maquina2(rw)
/home/paracompartir4       *.maquinas.tux(ro)
```

Una vez que hayan terminado de configurar el archivo **/etc/exports**, tienen que actualizar la base de datos de exportaciones. Esto pueden hacerlo de forma sencilla con el comando **exportfs -r**. El comando **exportfs** incluso nos permite exportar un directorio directamente desde la línea de comandos, sin alterar el archivo **/etc/exports** (¿será útil esto?). Veamos un ejemplo:

```
exportfs host:/home/paracompartir -o rw
```

Como pueden ver, la sintaxis de este comando es muy sencilla. Se definen el host y el directorio por compartir, y luego, con el parámetro **-o**, se definen las opciones necesarias.

Cómo montar directorios exportados

Ya configuramos totalmente el servidor y lo tenemos a la espera de peticiones de clientes en una red. Es hora de configurar el lado del cliente.
Para acceder a un directorio compartido en un servidor NFS, primero hay que montarlo. Este proceso no difiere en lo más mínimo de la operación de montado que usamos

comúnmente para acceder a disquetes, unidades de CD-ROM u otras particiones. Existe un comando que nos permite ver todos los directorios exportados que posee un servidor:

```
showmount -exports host
```

Aquí **host** puede ser el nombre de una de las máquinas de la red, o su dirección IP. Una vez que hayamos encontrado el directorio al cual queremos acceder, sólo nos resta montarlo con el clásico comando **mount**:

```
mount host:/directorio_exportado /puntodemontaje
```

Un ejemplo válido para los casos que mostramos en la sección anterior es el siguiente:

```
mount servidor:/home/paracompartir1 /mnt/minfs
```

Obviamente, **/mnt/minfs** es un directorio especialmente creado para montar los directorios de archivos exportados en servidores NFS.
Una vez que hayan terminado de usar los archivos en el servidor, recuerden que deben desmontar el directorio. Para esto, sólo hay que teclear:

```
umount /mnt/minfs
```

Si están usando NFS en un sistema con varias estaciones de trabajo, será necesario informar a todos los operarios sobre la utilización de este servicio. En algunos casos, hacer un script que monte automáticamente el sistema exportado de forma remota al inicio y desmonte todo cuando se apague el sistema es una buena idea.

CUESTIÓN DE PERMISOS

Recuerden que bajo este sistema, juegan de manera primordial los permisos de los archivos y directorios. Por eso, tengan muy en cuenta a quién le van a otorgar acceso, y tengan muy en claro cómo usar el comando chmod.

DIRECTORIOS OCUPADOS...

Desde el lado del cliente, recuerden desocupar el directorio NFS antes de desmontarlo. Esto es, salir de cualquier programa que pueda estar usándolo (por ejemplo, un administrador de archivos).

Usar NFS con cautela

Es absolutamente necesario que en entornos de red, el sistema NFS sea usado con mucha responsabilidad. Cuantos más directorios compartan, más posibilidades de problemas habrá (ataques externos, pérdidas de datos, etc.). Lo recomendable es tener un único directorio de archivos compartidos, en un lugar seguro, en una partición segura (como **/home**). De esta forma se logra aislar completamente los archivos exportados de los archivos locales del sistema.
Por último, es recomendable que cuando no utilicen el servicio de NFS, lo desactiven. Esto pueden hacerlo de varias formas, como usar la herramienta de manejo de servicios que venga incluida en su distribución, o ejecutando **/etc/rc.d/init.d/nfs stop**. Esto reducirá las posibilidades de ataques a nuestro sistema por este medio.

En resumen...

El sistema NFS es una interesante forma de compartir archivos y directorios en una red. Es interesante porque los clientes "montan" directamente el directorio exportado en el servidor y, de este modo, el acceso a los datos remotos es totalmente transparente para el sistema y para las aplicaciones que lo estén usando.
Para configurar un servidor NFS, simplemente hay que definir qué directorios queremos exportar en el archivo /etc/**exports**. En este archivo también se define qué permisos de acceso se otorgarán a los clientes que intenten acceder a los datos contenidos en ese directorio.
Por último, hay que tener siempre bien en cuenta el tema de los permisos de acceso a los archivos contenidos en directorios exportados, los cuales se manejan con el comando **chmod** (analizado en los primeros capítulos de este libro).
Ya hemos visto esta forma nativa de UNIX de compartir archivos. En los próximos capítulos, veremos otros métodos que nos permitirán compartir datos entre distintas plataformas de manera rápida y sencilla.

LA BIBLIA DE

Capítulo 11

Interconexión en red con Windows™

El sistema Samba nos permite compartir archivos entre sistemas Windows y GNU/Linux.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

Instalación y configuración de Samba

Es posible tener una red con máquinas Windows y máquinas GNU/Linux interconectadas. Esto es, las máquinas podrían enviarse y recibir archivos (compartiendo recursos) y, al mismo tiempo, compartir impresoras. Todo esto es posible gracias a Samba, uno de los proyectos de Software Libre más importantes del momento.
Samba generalmente viene instalado en casi todas las distribuciones de GNU/Linux; pueden comprobar esto buscando la existencia de **/etc/smb.conf**, su archivo de configuración. Si no lo tienen instalado, entonces sólo deben recurrir al CD de la distribución que están instalando y bajar todos los RPM, DEB o TGZ que tengan el nombre **samba**[*]. Una vez hecho esto, podemos proceder a configurar el sistema.
Como mencionamos anteriormente, Samba almacena toda su configuración en el archivo **/etc/smb.conf**. Luce así:

```
# Samba config file created using SWAT
# from localhost.localdomain (127.0.0.1)
# Date: 2002/07/11 17:09:49

# Global parameters
[global]
workgroup = tux
netbios name = LINUX
server string = Samba Server %v
interfaces = eth0
log file = /var/log/samba/log.%m
max log size = 50
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
printcap name = lpstat
dns proxy = No
printing = cups
print command = /usr/bin/lp -d%p -oraw %s; rm %s
lpq command = /usr/bin/lpstat -o%p
lprm command = /usr/bin/cancel %p-%j
queuepause command = /usr/bin/disable %p
queueresume command = /usr/bin/enable %p
encrypt passwords = yes
```

```
[homes]
comment = Home Directories
guest ok = Yes

[home]
path = /home
guest ok = Yes



[netlogon]
comment = Network Logon Service
path = /home
guest ok = Yes

[printers]
comment = All Printers
path = /var/spool/samba
create mask = 0700
guest ok = Yes
printable = Yes
print command = lpr-cups -P %p -o raw %s -r # using client side printer dri-
vers.
lpq command = lpstat -o %p
lprm command = cancel %p-%j

[/home]
comment = Personales
path = /tmp
guest ok = Yes
```

En este archivo definimos absolutamente todo. Desde los parámetros generales del funcionamiento del servidor, hasta los directorios e impresoras que compartiremos.

El sistema SWAT

Samba incluye una interfaz gráfica vía Web para configurar el archivo /etc/smb.conf. Para acceder a ésta, diríjanse a la dirección 127.0.0.1:901 en su navegador (asegúrense de que Samba esté corriendo). El sistema pedirá usuario y contraseña. Utilicen la cuenta de root si quieren tener acceso total a todas las opciones de configuración.

Echemos un vistazo a la sección **[global]**. En esta sección hay algunas cosas esenciales que debemos definir si queremos que nuestro servidor funcione correctamente. Éstas son:

- **workgroup**: aquí definimos el nombre de grupo de trabajo de la red Windows. Asegúrense de que sea exactamente igual que el que está definido en las computadoras que poseen el sistema operativo Windows. En caso contrario, será imposible establecer una comunicación entre estas computadoras.

- **server string**: en esta sección podemos definir una descripción de nuestro servidor, la cual será mostrada en la sección de información del Explorador de Windows.

- **netbios name**: aquí definimos el nombre de nuestra estación en la red. Éste será el nombre que los demás usuarios de la red verán desde sus computadoras.

- **interface**: qué interfaz de red se utilizará para conectarse por el protocolo de Samba. Por defecto, se utilizará **eth0**.

- **encrypt passwords**: esta opción habrá que definirla como "yes" si utilizan sistemas Windows desde la versión 98 en adelante (incluidas las versiones NT), ya que las claves en estos sistemas están encriptadas y no podrán ser reconocidas por Samba si el valor de esta opción es "no", o si directamente esta opción no está definida.

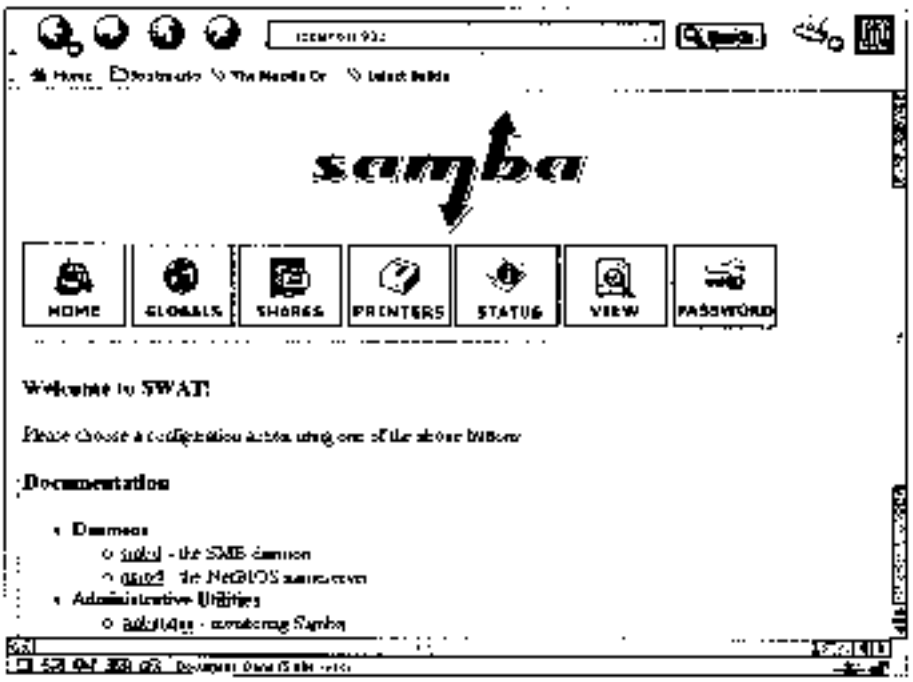


Figura 1. SWAT es un sistema que, por medio del navegador, nos permite configurar el archivo /etc/smb.conf de forma rápida y sencilla.

Una vez que están definidas estas opciones importantes, se puede pasar a la sección de compartimiento de recursos. Cada recurso comienza con una etiqueta descriptiva de tipo **[recurso]**. Para crear un nuevo recurso, será necesario crear una nueva etiqueta y definir algunas variables importantes. Pueden usar el siguiente modelo:

```
# Compartimiento de un directorio publico
[publico]
comment = Esta es un directorio publico
path = /home/publico
writable = no
public = yes
```

En la línea **comment** definimos el comentario que será visualizado en el Explorador de Windows. Luego, en la línea **path** definimos el directorio que vamos a compartir. Por último, no permitimos que nadie escriba nada allí y lo establecemos como directorio público.
Podemos hacer que a nuestro directorio pueda acceder cualquier usuario sin necesidad de ingresar su usuario y su contraseña (acceso de visitantes). Esto lo hacemos simplemente agregando la siguiente opción al final de la sección del directorio compartido:

```
guest ok = yes
```

Además, podemos definir desde qué direcciones IP pueden acceder a ese directorio compartido haciendo uso de la directiva **hosts allow** del siguiente modo:

```
hosts allow = [direccion IP aceptada] EXCEPT [direcciones IP no aceptadas]
```

Las direcciones IP pueden ser completas, o se pueden definir sólo algunos números; por ejemplo:

```
hosts allow = 10.0.
```

Permitirá el acceso desde todas las computadoras que tengan primeros números de dirección IP 10.0.

Una vez que hayan terminado de configurar este archivo, pueden verificar que la configuración sea correcta con el comando **testparm**.

```
[root@maquina2 /root]# testparm
Load smb config files from /etc/smb.conf
Processing section "[homes]"
Processing section "[home]"
Processing section "[netlogon]"
Processing section "[printers]"
Processing section "[/home]"
Loaded services file OK.
Press enter to see a dump of your service definitions
```

El sistema mostrará el estado de todas las variables de configuración. Ahora podemos comenzar a usar nuestro sistema Samba.

Encendiendo el servidor

Es momento de encender nuestro servidor Samba. Primero vamos a asegurarnos de que no esté corriendo en este mismo momento. Si así lo está, deberemos reiniciarlo para que se actualice con las opciones que definimos en el archivo de configuración. Para comprobar esto, simplemente pedimos un listado de procesos:

```
[root@maquina2 /root]# ps ax|grep smbd
1541 pts/2 S 0:00 grep smbd
[root@maquina2 /root]#
```

Obviamente, Samba no está funcionando. Vamos a iniciarlo dirigiéndonos primero al directorio **/usr/rc.d/init.d** y tecleando el siguiente comando:

```
[root@maquina2 init.d]# ./smb start
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
```

Listo. Pediremos nuevamente un listado de procesos y veremos que el demonio **smbd** ya está funcionando en nuestro sistema.

```
[root@maquina2 init.d]# ps ax|grep smb
1553 ?  S  0:00 smbd -D
1570 pts/2  S  0:00 grep smb
[root@maquina2 init.d]#
```

Cabe destacar que en algunas distribuciones puede que no aparezca el archivo **/etc/rc.d/init.d/smb**. En estos casos, podemos iniciar el demonio Samba directamente utilizando su comando nativo:

```
smbd -D
```

Ahora ya podemos comenzar a usar el sistema Samba.

Accediendo a servidores Windows

Para acceder a un directorio compartido en un servidor Windows tenemos dos herramientas nativas de Samba. La primera de ellas es **smbclient**, un sistema similar al FTP. La segunda consiste en el agregado de soporte al comando **mount** para poder montar directorios compartidos. Analicemos las dos opciones.

Smbclient es un comando de modo texto que nos permite comunicarnos con un servidor Windows, y subir y bajar archivos como si se tratara de un servidor FTP. La forma principal del comando **smbclient** es la siguiente:

```
smbclient //servidor/directorio_compartido -U [usuario] -W [grupo_de_trabajo]
```

NO TODO ES SMBCLIENT

Smbclient es la herramienta que nos provee Samba para manejarnos desde las consolas con otros servidores. De todas formas, existen muchas herramientas en Internet más gráficas y fáciles de usar que realizan la misma tarea. Es cuestión de buscar y seleccionar la que más se adecue a sus necesidades.

Cabe destacar que el nombre de usuario debe corresponder a un usuario existente en el servidor Windows. La clave que nos pedirá será la definida en el servidor Windows. Veamos un ejemplo concreto:

```
[root@maquina2 /root]# smbclient //servidor/vf2
added interface ip=10.0.0.2 bcast=10.255.255.255 nmask=255.0.0.0
Got a positive name query response from 10.0.0.3 ( 10.0.0.3 )
Password:
smb: \> dir
.  D  0  Fri Sep  8 15:27:28 2000
..  D  0  Fri Sep  8 15:27:28 2000
AKI_FACE.BIN  A  153600  Wed Feb 12 18:56:18 1997
AKIFACE8.BIN  A  76800  Thu Feb 13 11:19:40 1997
ARROW8.BIN  A  1152  Tue Feb  4 17:57:30 1997
ARROW.BIN  A  2304  Mon Mar 31 09:57:16 1997
BOX.BIN  A  91986  Tue Oct  8 09:25:42 1996
vf2.QID  AH  6628  Wed Feb 13 22:35:54 200

38831 blocks of size 131072. 16809 blocks available
smb: \> mget RUK.TXT
Get file RUK.TXT? y
getting file RUK.TXT of size 5516 as RUK.TXT (185.748 kb/s) (average 185.749
kb/s)
smb: \> quit
```

Como pueden observar, los comandos utilizados para visualizar información y obtener archivos son los mismos que los del protocolo FTP.
La otra forma de acceder a datos almacenados en servidores Windows es montando directamente esos recursos compartidos. De este modo, estaríamos trabajando casi de la misma manera que con NFS (acceso transparente a datos remotamente alojados), pero con un protocolo que nos permite comunicarnos entre sistemas Windows y Linux. Y lo mejor de todo es que sólo hay que usar los comandos del sistema que ya contienen el soporte de Samba incluido. Esto lo hacemos simplemente usando el comando **mount** del siguiente modo:

```
mount //servidor/directorio /punto_de_montaje -U [usuario]
```

El sistema preguntará por la correspondiente contraseña, y montará el directorio en el punto de montaje definido. Ya pueden usarlo como si de un directorio local se tratara; nuevamente, tengan en cuenta el tema de los permisos de acceso a los

archivos contenidos en estos directorios remotamente alojados. Una vez que hayamos concluido, podremos desconectarnos del servidor Windows desmontando directamente el directorio:

```
umount /punto_de_montaje
```

Accediendo a recursos compartidos en máquinas GNU/Linux

Para acceder a recursos compartidos en máquinas GNU/Linux desde Windows, será necesario que el nombre de usuario y la contraseña del usuario de Windows estén definidos en el servidor GNU/Linux. Por ejemplo, si el usuario de Windows utiliza el nombre de usuario "facundo" y la contraseña "12345", entonces deberán existir un usuario y una contraseña igualmente definidos en el servidor GNU/Linux. Éstos son los pasos por seguir para registrar el usuario:

Registrar un usuario — PASO A PASO

1. Creen el perfil del nuevo usuario con el comando **adduser [nombre_de_usuario]**.
2. Asignen una clave a ese usuario con el comando **passwd [usuario]**.
3. Creen el perfil del nuevo usuario en la base de datos de Samba con el comando **smbadduser [usuario]**.
4. Asignen la misma clave que el usuario tiene en Windows con el comando **smbpasswd [usuario]**.

Ahora, cuando el usuario de Windows acceda al entorno de red, verá el nombre de su servidor GNU/Linux en el listado de servidores (parámetro **netbios name** de **/etc/smb.conf**). Cuando haga clic allí, podrá obtener un listado de todos los directorios compartidos, y con sólo hacer doble clic en alguno de ellos, podrá acceder a la información que contiene.
Es un sistema realmente fácil de configurar y sumamente útil para interconectar datos entre máquinas que contengan diferentes sistemas operativos, pero que soporten el protocolo de datos SMB.

En resumen...

Samba es uno de los proyectos más grandes y exitosos del mundo del Software Libre. Esto se debe principalmente a dos razones: la primera es que funciona realmente bien, y permite que los usuarios de diferentes sistemas operativos (no sólo Windows y GNU/Linux, sino también Mac OS X, por ejemplo) comuniquen sus datos de forma totalmente transparente. La segunda razón es que se trata de un sistema muy fácil de configurar y administrar. Si bien el archivo de configuración de Samba está sumamente comentado y bien estructurado (lo que facilita muchísimo su edición), a algún usuario se le puede complicar el proceso de configuración. Por eso existen muchas herramientas gráficas que permiten configurar el sistema con sólo hacer un par de clics. En este capítulo les mostramos uno de estos sistemas (SWAT), el cual está incluido dentro de la misma distribución de Samba y permite configurar todo el servidor desde un navegador.

En conclusión, se trata de un sistema sumamente avanzado, rápido y fácil de configurar, que le ha hecho la vida más sencilla a millones de usuarios de GNU/Linux y Windows de todo el mundo.

LA BIBLIA DE

Capítulo 12

Firewalls y proxies

El sistema IPTables, incluido en la serie 2.4 del kernel Linux, nos permite definir firewalls y proxies de forma sumamente práctica.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

¿Qué es un firewall?

En este capítulo vamos a ponernos un poco más técnicos aún, y hablaremos de algunas herramientas avanzadas de red que permitirán hacer mucho, con muy poco esfuerzo. El núcleo Linux es realmente un arma suiza. Posee de todo, incluidos un sistema de firewall (cortafuego) y un proxy transparente. ¿Por qué hablamos de estos dos términos conjuntamente? Porque son gestionados por la misma herramienta (que forma parte del propio núcleo).

Un firewall es una herramienta de seguridad en redes muy utilizada. En realidad, una red que esté conectada a Internet y no utilice un firewall es una red totalmente insegura.

La función principal del firewall es la de arbitrar la entrada, la salida y el paso de paquetes entre una red y otra. Por ejemplo, podemos tener una red de cuatro máquinas conectadas a Internet, en donde una sola de ellas es el servidor web. Desde Internet, es posible acceder a todos los demás servicios (siempre y cuando éstos estén levantados). Mediante el uso de un firewall, podemos hacer que los que están fuera de nuestra red interna (Internet) sólo puedan acceder al puerto del servidor web. Los demás serán totalmente inaccesibles.

La cantidad de cosas que se pueden hacer con un firewall es impresionante; podemos limitar el paso de datos entre máquinas locales, podemos evitar que nos envíen paquetes **icmp** (los del comando **ping**) y muchas acciones más. Veamos cómo es su uso.

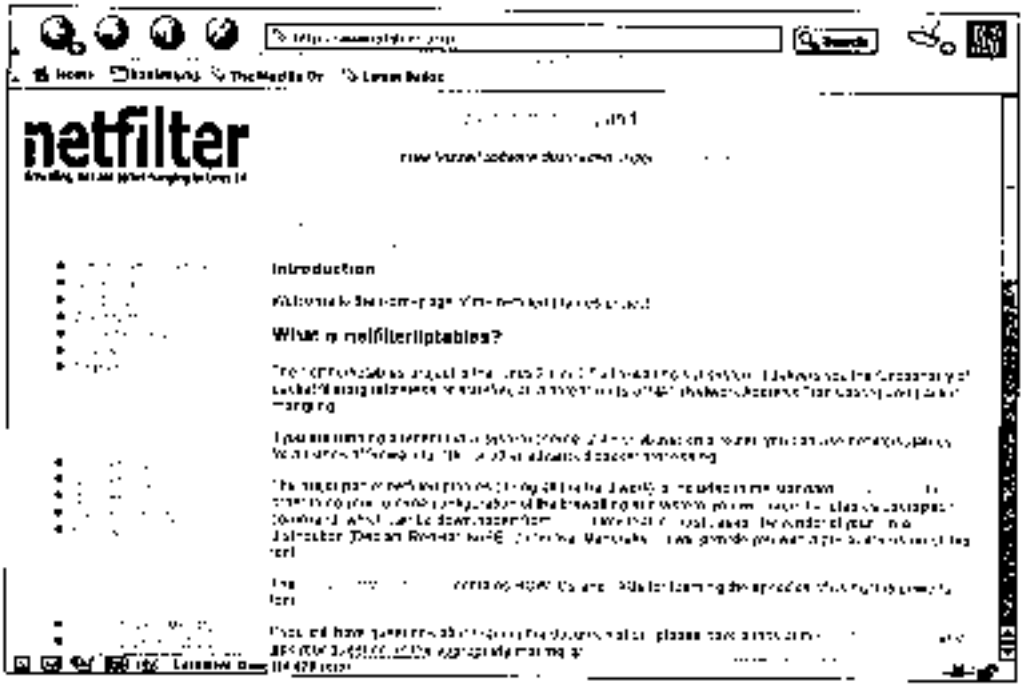


Figura 1. El sistema de filtrado de paquetes ***iptables*** *posee su propio sitio oficial en* ***www.iptables.org****.*

El comando iptables

Iptables es el sistema de firewalling en la serie 2.4 de kernels Linux. Antes teníamos **ipchains** (en la serie 2.2) e **ipfwadmin** (en la serie 2.0). Todos estos comandos poseen parámetros similares, así que no se asusten si tienen otra versión.
Iptables se maneja, básicamente, por la definición de *qué puede entrar, qué puede pasar* y *qué puede salir*. Para montar un firewall, nos importará definir qué es lo que puede entrar y qué es lo que puede salir desde nuestra máquina. Esto lo hacemos definiendo reglas de entrada y salida. Para ver todas las reglas definidas, tipeamos el comando **iptables -L**. Obviamente, si lo tipean ahora, verán que no hay reglas definidas.
La forma básica del comando **iptables** es la siguiente:

```
iptables -[A/D] [REGLA] [OPCIONES]
```

Los parámetros se estructuran en tres grupos. El primero es para crear una nueva regla (**A**) o para borrar una regla existente (**D**). Cabe destacar que si quieren borrar una regla previamente creada, deberán definir todos los demás parámetros exactamente de la misma forma en que fueron creados. Muchas veces simplemente se puede ir al historial de Bash con las flechas del cursor, encontrar el comando con el cual se creó esa regla, y luego, cambiar la **A** por una **D**.
En la sección **regla** se definen *origen* y *destino*, y luego, básicamente, se determina qué hacer con esos paquetes que entran o que salen de nuestro sistema.

Instalando un firewall

Si queremos que todo lo que entre desde la IP 200.100.122.200 sea rechazado, usaremos el siguiente comando:

```
iptables -A INPUT -s 200.100.122.200 -j DROP
```

La opción **-s** define la fuente del paquete. La opción **-j** puede llevar dos parámetros: **ACCEPT** y **DROP**. El primero permitirá el paso de los paquetes, el segundo no.
Si queremos anular la salida de paquetes hacia una cierta IP definida:

```
iptables -A OUTPUT -d 200.100.122.200 -j DROP
```

Ahora ningún paquete que vaya a la dirección 200.100.122.200 podrá salir de nuestra máquina. También podemos rechazar los paquetes que entren en nuestro sistema por el protocolo **icmp** (el del ping).

```
iptables -A INPUT -p icmp -j DROP
```

Con este comando, nadie podrá hacer un ping a nuestro sistema.
Iptables posee otras opciones muy interesantes; éstas son algunas de ellas:

OPCIÓN	DESCRIPCIÓN
-p [protocolo]	Trabajar sólo con los paquetes de un protocolo definido.
—source-port [puerto]	Trabajar sólo con los paquetes que vienen de un puerto definido.
—destination-port [puerto]	Trabajar sólo con los paquetes que salen por un puerto definido.
-i [interfaz de red]	Trabajar sólo con la interfaz de red definida.

Existen muchas otras posibilidades. Es recomendable que se enteren de ellas leyendo la documentación oficial de **iptables** o tecleando **man iptables**.

Instalando un proxy

Muchas veces surge la necesidad de tener muchas computadoras conectadas en red, y que, al mismo tiempo, todas ellas tengan acceso a la gran Red de redes (Internet). Generalmente, una sola computadora es la que obtiene el acceso a Internet y, consecuentemente, la que obtiene la dirección IP real de Internet. Las demás computadoras de la red deben tener una IP pública y no tienen salida directa. Tampoco entrada, claro.
La solución para estos sistemas es usar NAT (*Network Address Translation*, o traducción de direcciones de red). En el mundo de GNU/Linux, esto es comúnmente llamado IP Masquerading (enmascaramiento de IP) o Proxy transparente.

DOCUMENTACIÓN

En el Indlug (www.indlug.org) existe un HOWTO que se dedica a explicarnos cómo instalar un proxy basado en IP Masquerading. Posee numerosos ejemplos y una interesante explicación teórica de cómo funciona el sistema.

PROXIES CON CACHE

En este capítulo abordamos el tema de configuración de proxies usando iptables, el cual no provee sistema de caché. Si necesitan esta funcionalidad, les recomendamos encarar el proyecto utilizando algún sistema como SQUID.

¿Cómo funciona esto? Simple. Una sola computadora se conecta a Internet y tiene una dirección IP real. Las demás utilizan esta computadora como pasarela (gateway). Ahora, el servidor de gateway se encargará automáticamente de rutear los paquetes entre Internet y las computadoras de nuestra red.
Para llevar a cabo esta tarea, es necesario tener instalado un kernel con soporte para iptables en el servidor (generalmente, todos los kernels lo tienen). De todas formas, para asegurarnos de que todo lo que necesitamos esté activado, tipearemos el siguiente comando:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Una vez que hayamos hecho esto, podremos proceder a configurar una regla de iptables para que envíe todos los paquetes entrantes desde la red interna hacia Internet. Crearemos una regla para el paso de datos por nuestra computadora (el servidor).

```
iptables -A FORWARD -s 10.0.0.0/24 -d 0.0.0.0/0 -j MASQ -p tcp
```

Analicemos un poco cómo está funcionando esto. Con la opción **-A forward** estamos creando una regla de paso de paquetes. Luego definimos la dirección **10.0.0.0/24** como fuente de los paquetes. Obviamente, pueden poner cualquier dirección IP que usen en su red interna. El **/24** define el tipo de red que se utilice. Para saber qué valor usar, se emplea la siguiente tabla:

MÁSCARA	N	RED CLASE
255.0.0.0	8	A
255.255.0.0	16	B
255.255.255.0	24	C
255.255.255.255	32	PtP

Como destino, definimos la dirección **0.0.0.0/0**, o sea, cualquier destino. Por último, definimos como acción el enmascaramiento (**MASQ**) del protocolo TCP.

DIFERENCIAS ENTRE LOS NÚCLEOS

En la serie 2.0, el proxy/firewall se maneja con el sistema ipfwadmin. En la serie 2.2, con el sistema ipchains, y en la serie 2.4, con iptables. El uso de estos tres sistemas es muy similar, y sus parámetros son casi los mismos. Es sólo cuestión de pedir un man de cada comando para notar cuáles son las diferencias principales.

Lo único que resta hacer es, en cada estación de nuestra red interna, definir como gateway la dirección IP de nuestro servidor proxy. No importa qué sistema operativo estén usando en las estaciones, el proceso será totalmente transparente.
Cabe destacar que es posible combinar esto con los parámetros anteriores que vimos de firewalling, para poder obtener un proxy seguro y permitir el paso de paquetes sólo por un puerto definido o negar ciertos puertos. Hagan sus propias investigaciones.

En resumen...

El sistema de filtrado de paquetes incluido en el kernel Linux nos provee de dos herramientas sumamente necesarias a la hora de montar una red. La primera de ellas es la utilización de un firewall (cortafuegos) para filtrar todo lo que entra y sale de nuestra red y, de esta manera, aumentar su seguridad. La segunda es el montaje de un servidor proxy, el cual nos permite configurar una puerta de enlace para proveer de acceso a Internet a varias computadoras de una red, con sólo una IP real.
Por último, cabe destacar que, como se trata de un sistema incluido dentro del núcleo (no necesita herramientas adicionales), es posible tener una distribución de GNU/Linux que quepa en un disco floppy y se encargue de hacer todo el filtrado de paquetes en una red. Generalmente, las empresas usan computadoras viejas para esto (hablamos de sistemas 486, con poca memoria RAM y una distribución que es ejecutada desde un floppy, para no gastar en unidades de disco rígido).

LA BIBLIA DE

Capítulo 13

El sistema VNC

Ya hablamos de acceso remoto a través de terminales textuales. Ahora nos referiremos al control remoto de interfaces gráficas.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

Introducción

El sistema VNC (Virtual Network Computing) permite, básicamente, utilizar el escritorio de un sistema operativo de forma remota. Esto significa, que podemos controlar un sistema desde otra terminal de una red, o desde cualquier computadora de Internet. El funcionamiento de VNC es bastante sencillo. Básicamente, se ejecutan dos aplicaciones que permiten llevar a cabo el proceso de control remoto. Por un lado tenemos el servidor VNC, el cual es ejecutado en el sistema operativo que queremos controlar remotamente. Desde otra terminal, ejecutaremos un cliente VNC, que se encargará de establecer la conexión y mostrarnos en pantalla lo que supuestamente debería verse en el servidor.

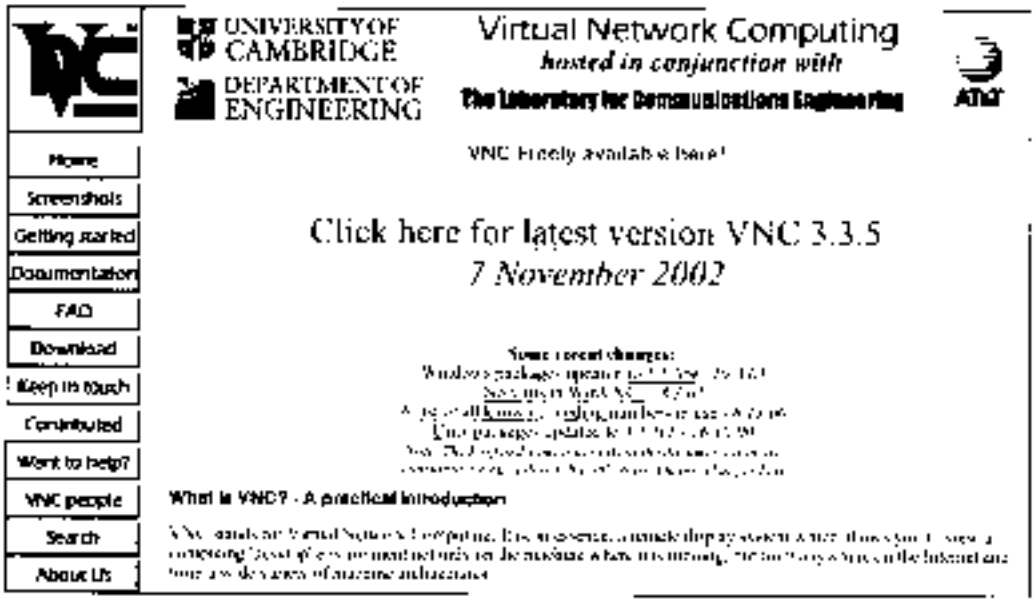


Figura 1. Actualmente, el sistema VNC está siendo desarrollado en conjunto por los laboratorios AT&T y el Departamento de Ingeniería de la Universidad de Cambridge.

Características de VNC

Éstas son algunas de las características que hacen de VNC un sistema muy interesante de utilizar:

Es multiplataforma

Existen servidores y clientes para casi todos los sistemas operativos que poseen un entorno gráfico. Lo mejor de todo es que no hay necesidad de que el cliente y el

servidor estén corriendo en el mismo sistema, sino que podemos tener, por ejemplo, un servidor corriendo en un sistema Windows, y el cliente puede estar ejecutándose desde un sistema GNU/Linux bajo Xwindow. Lo más interesante es que hay clientes de VNC para JAVA, lo que quiere decir que ustedes pueden estar en la China viendo la pantalla del monitor de una computadora que está en Nueva York, por medio del navegador.

No se almacena información en el cliente.

El cliente actúa sólo como visor del servidor. Ningún tipo de información del servidor se almacena en el cliente. Además, ustedes pueden estar trabajando en su escritorio, cerrar el cliente, y el escritorio quedará en el mismo estado. Cuando vuelvan a conectarse al servidor VNC, el escritorio estará exactamente igual que como lo dejaron, con las mismas aplicaciones y en el mismo estado.

Permite compartir conexiones.

Una sesión VNC puede ser compartida por muchos clientes de forma simultánea. Hay que tener cuidado con las condiciones de competencia que se dan en algunos sistemas operativos como Windows (ya hablaremos de eso más adelante).

Es compacto.

Tanto el cliente como el servidor ocupan muy pocos KBs y se ejecutan hasta en los sistemas más carentes de recursos. ¡Vean los beneficios! Ustedes pueden tener un gran servidor de aplicaciones y muchas terminales de bajo presupuesto (hardware limitado) que accedan al servidor y usen las aplicaciones desde allí. ¡Imaginense un sistema 486 con 16 MB mostrando KDE 3 con Mozilla y StarOffice en la pantalla!

Es Software Libre.

Por último, estamos hablando de una aplicación 100% Software Libre. Esto significa que su código fuente está a disposición de todo aquel que tenga interés en obtenerlo, así como se permite su libre distribución.

EL SERVIDOR DE WINDOWS

El servidor de VNC de Windows (WinVNC) es un tanto particular, ya que no soporta múltiples usuarios en diferentes sesiones. La gente de AT&T informa en la documentación de este protocolo que el servidor WinVNC se limita simplemente a mostrar el escritorio en una pantalla remota, desde donde también se podrá controlar el sistema. Si hay muchos usuarios, deberán "pelear" por mover el cursor del mouse.

¿Cómo funciona VNC?

Si bien no es un sistema muy "popular" entre los usuarios de GNU/Linux, es indispensable que el lector sepa que cuenta con una herramienta de gran utilidad, al alcance de la mano: VNC. El sistema VNC es un protocolo que, básicamente, funciona transfiriendo imágenes del servidor al cliente. El servidor se encarga de realizar todas las tareas importantes de la conexión; por consiguiente, los clientes son programas muy pequeños que sólo se limitan a recibir el mensaje correspondiente, que, generalmente, es del tipo "poner un pixel de n color en la posición x,y". Esto es lo que permite que haya tantos clientes de VNC para las diferentes plataformas y que sean tan livianos (corren en casi cualquier hardware desde 386 en adelante). Todo esto se ve beneficiado, además, por el hecho de que programar un cliente de VNC es una tarea muy sencilla. No así un servidor.

Lo primero que hay que hacer para llevar a cabo una sesión VNC es levantar un servidor VNC en un sistema operativo que esté soportado por este protocolo. En el servidor se definirá un usuario y una contraseña que serán los que usarán los clientes cada vez que quieran conectarse a él. Desde el lado del cliente, sólo se le pasa el número de IP del servidor y el número de sesión VNC a la cual se quiere conectar (como mencionamos anteriormente, es posible tener múltiples sesiones VNC en diferentes estados de forma simultánea), y ya podrán estar trabajando de forma remota sin problemas.

Obteniendo e instalando el paquete VNC

Para instalar VNC, hay que dirigirse al sitio oficial de VNC, ubicado en **www.uk.research.att.com/vnc/**. Allí, en la sección "downloads" encontrarán los diferentes paquetes para los distintos sistemas operativos. Nosotros nos encargaremos solamente de la versión para GNU/Linux, así que descarguen el paquete .tar.gz y descomprímanlo en algún directorio de su disco rígido.
Tengan en cuenta que el paquete ya viene compilado, por lo que no habrá que preocuparse por el tedioso proceso de compilación.

Una vez descomprimido, encontrarán un listado de directorios similar al siguiente:

```
[hfarena@alternarama vnc_x86_linux_2.0]$ ls
LICENCE.TXT  README  Vncviewer  Xvnc  classes  vncconnect  vncpasswd
vncserver  vncviewer
[hfarena@alternarama vnc_x86_linux_2.0]$
```


Como pueden ver, el paquete es bastante compacto, no hay muchos archivos. Esto simplifica muchísimo el proceso de instalación, del cual hablaremos ahora mismo.

Para instalar VNC, lo que tienen que hacer es copiar los binarios ubicados en este directorio a algún directorio de binarios que figure en la variable de entorno **$PATH**. Recordemos que los directorios que figuran en dicha variable contienen archivos binarios que pueden ser ejecutados desde cualquier punto del sistema de archivos. De forma que el siguiente ejemplo es válido para cualquier sistema GNU/Linux:

```
% cp vncviewer vncserver vncpasswd vncconnect Xvnc /usr/local/bin
```

Esto copiará los archivos definidos en el comando al directorio **/usr/local/bin**. Ahora ya estamos listos para ejecutar el servidor VNC.

Ejecución del servidor

Antes de encender nuestro primer servidor VNC, es necesario aclarar una cosa. El servidor puede ser ejecutado por el administrador del sistema, o por algún usuario común. Si es ejecutado por el administrador, entonces cualquiera que se conecte remotamente tendrá todos los privilegios del administrador (por eso, tengan mucho cuidado con esta modalidad de ejecución). Ahora, si ejecutamos el servidor desde una cuenta de usuario normal, entonces el usuario que se conecte remotamente tendrá los privilegios del usuario que ejecutó el servidor. Habiendo aclarado esto, levantaremos el servidor VNC con el comando **vncserver**.

```
[hfarena@alternarama hfarena]$ vncserver
You will require a password to access your desktops.

Password:xxxx
Verify:xxxx
```

```
New 'X' desktop is alternarama:1
Creating default startup script /home/hfarena/.vnc/xstartup
Starting applications specified in /home/hfarena/.vnc/xstartup
Log file is /home/hfarena/.vnc/alternarama:5.log
[hfarena@alternarama hfarena]$
```

Analicemos un poco esta salida. Lo que el sistema nos pide es una clave que tendrán que usar los clientes cada vez que quieran acceder a escritorio del usuario **hfarena** (quien ejecutó el servidor). Esta clave la pide únicamente, la primera vez que se ejecuta el servidor. Luego, queda almacenada en el sistema y, si queremos definir una nueva clave, será necesario borrar el archivo donde está definida.

Luego, el sistema nos indica que ha creado un nuevo escritorio X para el servidor, ubicado en **alternarama:1**. Alternarama es el nombre de host de la máquina que fue usada para hacer las pruebas de este libro, y el número 1 nos indica que es el único servidor VNC que hay ejecutándose en el sistema. Si otro usuario ejecuta otro servidor VNC, entonces la dirección será **alternarama:2**, y así sucesivamente. Lo importante es que entiendan que para hacer referencia a un servidor VNC desde un cliente, hay que respetar la siguiente nomenclatura:

```
[nombre_de_host/IP:[numero_de_servidor_VNC]
```

Las últimas tres líneas nos indican que se han creado dentro del directorio **$HOME/.vnc** los archivos necesarios para la ejecución del servidor, los cuales analizaremos más adelante.

Bueno, ya tenemos nuestro servidor instalado y en ejecución; es hora de probarlo desde otra estación.

El cliente VNC

Es momento de hablar un poco de los clientes VNC. Como mencionamos al principio de este capítulo, el cliente de VNC puede ser para otra plataforma, que no sea la misma que la del servidor. O sea, puede ejecutar un cliente VNC para Microsoft Windows, para acceder a un servidor localizado en un sistema GNU/Linux, y viceversa. Nosotros nos remitiremos a hablar solamente del cliente de VNC para GNU/Linux, el cual se invoca con el comando **vncviewer**.

VNCViewer es una aplicación de Xwindow que requiere muy pocas librerías para funcionar (en realidad, con que sencillamente esté instalado Xwindow, será más que suficiente). Para ejecutarlo, abran una terminal virtual de Xwindow y tipeen el comando **vncviewer**.

*Figura 2. Lo primero que nos pide VNC es que ingresemos la dirección del servidor al cual queremos acceder. Recuerden que se define bajo la nomenclatura **[host]:[numero]***

Una vez que hayamos ingresado la dirección del servidor, éste nos pedirá la clave para acceder.

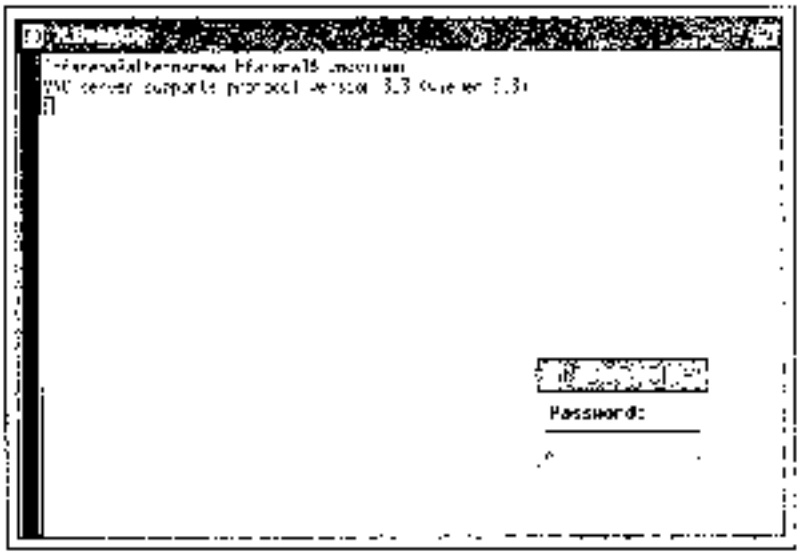


Figura 3. La clave que tienen que usar es la que definió el usuario que ejecutó el servidor VNC.

Listo. Aparecerá una ventana en pantalla con el escritorio remoto. Ya pueden usar el sistema, tal como si tuvieran sus manos posadas sobre el teclado y el mouse del servidor.

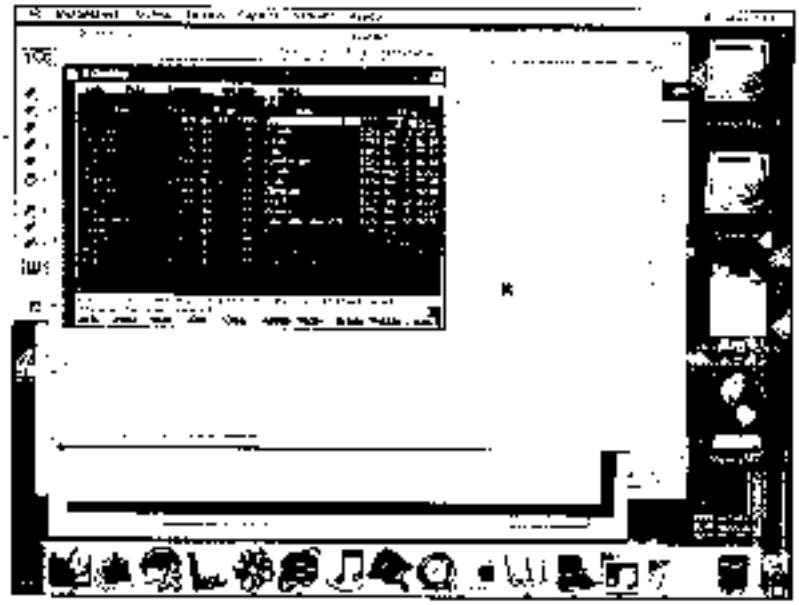


Figura 4. En esta pantalla podemos ver un cliente de VNC corriendo bajo el sistema operativo Mac OS X conectado a un servidor basado en un sistema GNU/Linux.

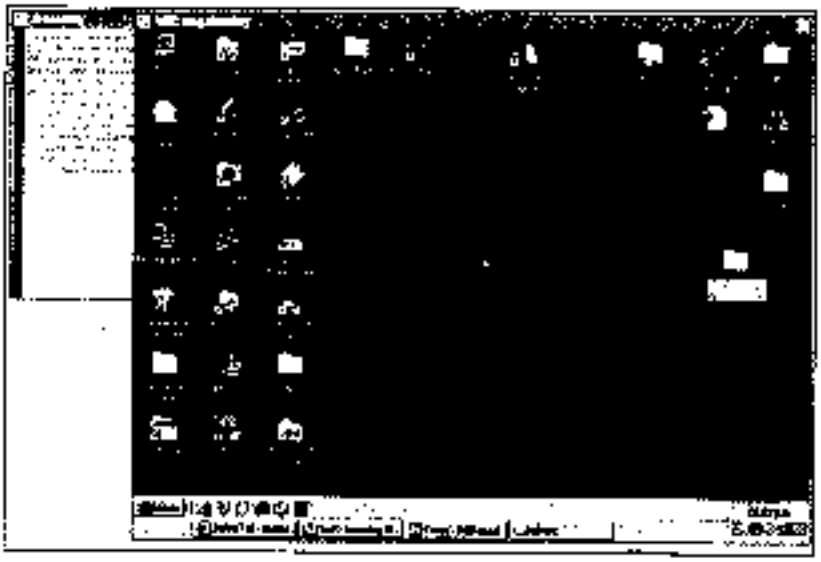


Figura 5. En esta otra, podemos ver un cliente de VNC siendo ejecutado bajo GNU/Linux, y accediendo a un servidor Windows (en realidad, el cliente de VNC de Linux está siendo controlado desde otro cliente de VNC que está corriendo sobre OS X).

Apagado del servidor

Cuando los clientes cierran la aplicación que están utilizando para acceder a nuestro servidor, el estado del sistema queda intacto. El servidor no se cierra y las aplicaciones que estaban abiertas siguen en el mismo estado, hasta que un nuevo usuario se

conecte. Por eso, si queremos cerrar el servidor, debemos hacerlo nosotros mismos (la combinación **ssh/VNC** es a veces muy útil, ya me entenderán por qué).
Para cerrar el servidor VNC, ejecuten el comando **vncserver** con los siguientes parámetros:

```
vncserver -kill :[número]
```

Aquí, **[número]** es el número de servidor VNC que queremos cerrar. Para nuestro ejemplo,

```
[hfarena@alternarama hfarena]$ vncserver -kill :5
Killing Xvnc process ID 1398
[hfarena@alternarama hfarena]$
```

Listo. El servidor ya está cerrado. Es importante que entiendan que con este comando se cierran el servidor X, el servidor VNC y todas las aplicaciones que estaban siendo ejecutadas en esa sesión.

VNC, a modo mío

Hasta este momento, ya sabemos cómo levantar un servidor VNC, y ya podemos acceder a él desde un cliente VNC, que puede estar ejecutándose en cualquier otro sistema operativo que no sea GNU/Linux. Pero surge un problema: cada vez que un servidor se ejecuta, ejecuta los programas que él quiere (o sea, cualquier administrador de ventanas muy básico y una terminal para que el usuario pueda lanzar sus propios programas). Entonces, lo que queremos hacer ahora es que, cuando se levanta el servidor VNC, se ejecute un programa X que nosotros definimos.

Esto se maneja desde el archivo **$HOME/.vnc/xstartup**, que no es más que un script de Bash que se ejecuta en el momento de lanzar el servidor. Veamos el contenido actual de este archivo:

```
[hfarena@alternarama .vnc]$ cat xstartup
#!/bin/sh

xrdb $HOME/.Xresources
```

```
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
[hfarena@alternarama .vnc]$
```

Las primeras líneas de este archivo (las primeras dos, en realidad) preparan el entorno gráfico. Las últimas dos (que terminan con un símbolo "&") definen que programa se ejecutará apenas el servidor esté levantado. Como podemos ver, se ejecuta una **xterm** (terminal virtual) y el **twm** que no es más que un administrador de ventanas muy sencillo. Bueno, si queremos cambiar el administrador de ventanas, ésta es la línea que debemos cambiar. Si no queremos ningún administrador de ventanas y sólo queremos que el sistema ejecute un programa definido (muy útil en terminales públicas), entonces solo tenemos que poner la ruta completa al ejecutable y terminar la línea con un símbolo "&", para que el script continúe su ejecución y no se quede parado en ese comando.

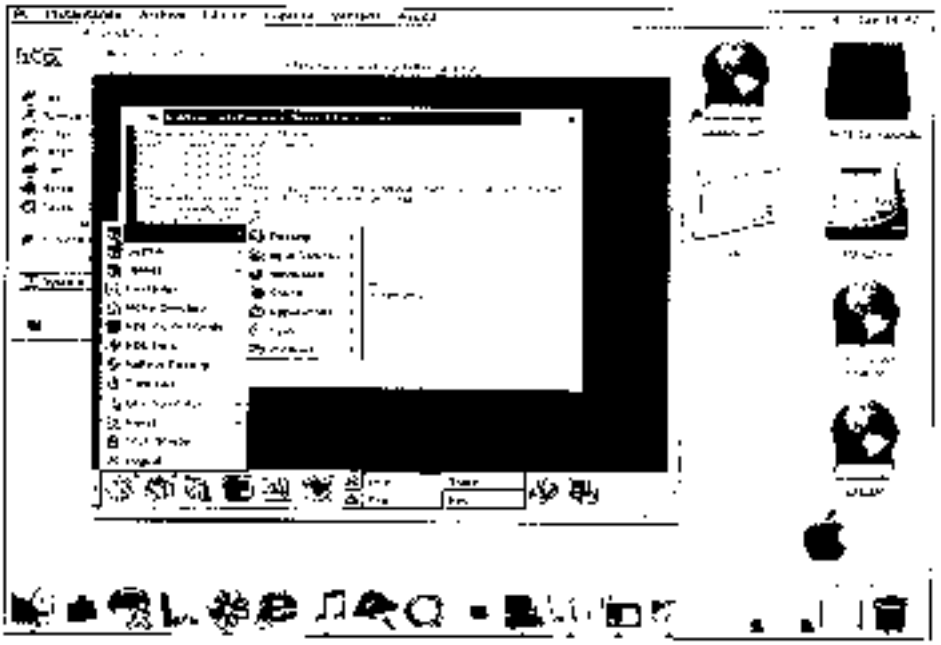


Figura 6. En esta ocasión hemos cambiado el administrador de ventanas ***twm*** *que venía preconfigurado por el popular KDE (versión 1).*

Otro archivo importante de este directorio es **passwd**, el cual contiene la clave definida por el administrador en el momento de ejecutar el servidor por primera vez. Si queremos cambiar la clave, sólo debemos eliminar ese archivo.

Por último, en este directorio encontraremos archivos con extensión .log, que registran toda la actividad del servidor. Veamos el contenido de uno de ellos (en el caso del ejemplo, **alternarama:5.log**).

```
[hfarena@alternarama .vnc]$ cat alternarama:5.log
11/12/02 23:21:24 Xvnc version 3.3.3r2
11/12/02 23:21:24 Copyright (C) AT&T Laboratories Cambridge.
11/12/02 23:21:24 All Rights Reserved.
11/12/02 23:21:24 See http://www.uk.research.att.com/vnc for information on
VNC
11/12/02 23:21:24 Desktop name 'X' (alternarama:5)
11/12/02 23:21:24 Protocol version supported 3.3
11/12/02 23:21:24 Listening for VNC connections on TCP port 5905
11/12/02 23:21:24 Listening for HTTP connections on TCP port 5805
11/12/02 23:21:24 URL http://alternarama:5805
Font directory '/usr/X11R6/lib/X11/fonts/100dpi/' not found - ignoring
xrdb: No such file or directory
xrdb: can't open file '/home/hfarena/.Xresources'

11/12/02 23:21:36 Got connection from client 10.0.0.1
11/12/02 23:21:36 Protocol version 3.3
11/12/02 23:21:36 Pixel format for client 10.0.0.1:
11/12/02 23:21:36 8 bpp, depth 8
11/12/02 23:21:36 true colour: max r 7 g 7 b 3, shift r 0 g 3 b 6
11/12/02 23:21:36 no translation needed
11/12/02 23:21:36 Using hextile encoding for client 10.0.0.1
11/12/02 23:24:44 Client 10.0.0.1 gone
11/12/02 23:24:44 Statistics:
11/12/02 23:24:44 key events received 48, pointer events 903
11/12/02 23:24:44 framebuffer updates 2035, rectangles 11952, bytes 1111042
11/12/02 23:24:44 copyRect rectangles 61, bytes 976
11/12/02 23:24:44 hextile rectangles 11891, bytes 1110066
11/12/02 23:24:44 raw bytes equivalent 8273505, compression ratio 7.453165
X connection to :5.0 broken (explicit kill or server shutdown).
[hfarena@alternarama .vnc]$
```

Como pueden apreciar, nos informa de todas las conexiones que se han realizado (incluidas las direcciones IP) y las desconexiones, y ofrece fecha y hora de cada una de estas actividades. Es recomendable que revisen periódicamente estos archivos, ya que nos pueden informar de cualquier anomalía posible en el sistema.

En resumen...

El servicio VNC nos provee de una manera sencilla de manejar remotamente una computadora, a través de una terminal gráfica. El estupendo trabajo que está haciendo el laboratorio de Ingeniería de la Universidad de Cambridge en sociedad con AT&T ha permitido que el protocolo sea todo un éxito, debido, entre otras cosas, a la independencia total del sistema operativo en el cual se está ejecutando.
Lo mejor de todo esto es que se trata de un sistema totalmente Software Libre, distribuido bajo la Licencia Pública General. Tengan en cuenta que, cuando bajan los paquetes oficiales precompilados, el código fuente no viene incluido, pero pueden descargarlo desde el mismo sitio de donde obtuvieron el paquete listo para usar.
Muchos dicen que el camino del protocolo VNC recién comienza, y ya hay muchos proyectos que lo están mejorando día a día (existen formas de hacer correr VNC sobre un protocolo encriptado, por ejemplo). Es sólo una cuestión de tiempo hasta que obtengamos un sistema seguro, simple de usar y sumamente rápido. Por el momento es totalmente utilizable, y muchas redes del mundo lo emplean para simplificar ciertas tareas, o ahorrar costos a la hora de montar un sistema de control remoto a través de interfaces gráficas. Definitivamente, VNC es una posibilidad más para tener en cuenta a la hora de montar servidores de red.

LA BIBLIA DE

Capítulo 14

Clusters Beowulf

Existe un método para hacer que muchas computadoras de poca potencia trabajen en conjunto para formar un supersistema de cómputo.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

¿Qué son los clusters?

Las supercomputadoras son usadas en bancos, universidades y centros de investigación para realizar complejos cálculos y obtener rápidos resultados. Seguramente habrán visto en fotos esas grandes máquinas que ocupan cuartos completos y que permiten ejecutar miles de millones de operaciones por segundo. Esas "heladeras" poseen varios procesadores, inmensas cantidades de memoria y un espacio de almacenamiento mucho más grande de lo que puedan imaginar. Todos estos recursos pueden estar dispuestos de dos formas: o están centralizados (todo dentro de una gran caja) o están distribuidos. Esta última modalidad es la que nos interesa a nosotros.
Vamos a decir que los clusters son computadoras potentes que funcionan gracias a un sistema que permite compartir los recursos de varios subsistemas. Podemos tener dos o más computadoras interconectadas entre si por una red y hacer que compartan libremente sus recursos. Y cuando hablamos de recursos, no sólo nos referimos a archivos e impresoras: los clusters permiten que las computadoras compartan sus procesadores y sus memorias RAM.
Se estarán preguntando, entonces: si conectamos dos computadoras en cluster... ¿podemos obtener una sola con el doble de potencia? En cierta forma, si. La ganancia de rendimiento no es de exactamente el doble, pero es muy superior a la de cada unidad del cluster. En muchas universidades se usan clusters de 40 o 50 computadoras de "bajo" rendimiento, como 486 y Pentiums de los primeros. Estas máquinas, trabajando en conjunto, ofrecen un supersistema de cómputo muy eficaz, cuyo costo es significativamente menor que el de una supercomputadora cuyos recursos están centralizados en una gran caja.

Clusters en GNU/Linux

Nuestro querido sistema operativo nos permite hacer esto de forma supersencilla. Todo lo que necesitamos son un par de máquinas conectadas entre si (una red convencional es suficiente) y el software requerido.
Mosix es un proyecto que se dedica a desarrollar el programa necesario para cumplir con esta función. Instalado en cada una de las máquinas, Mosix permite que trabajen todas en conjunto, compartiendo memorias y capacidades de procesamiento. Trabaja de forma "inteligente"; por ejemplo, cuando una máquina está al borde de ser saturada por los procesos, el sistema busca otra que esté ociosa y le "saca" algunos recursos para otorgárselos a la máquina saturada. Este sistema es totalmente transparente para los usuarios, y ellos no tienen que saber absolutamente nada de clusters. Una vez instalado, todo funciona de forma independiente.

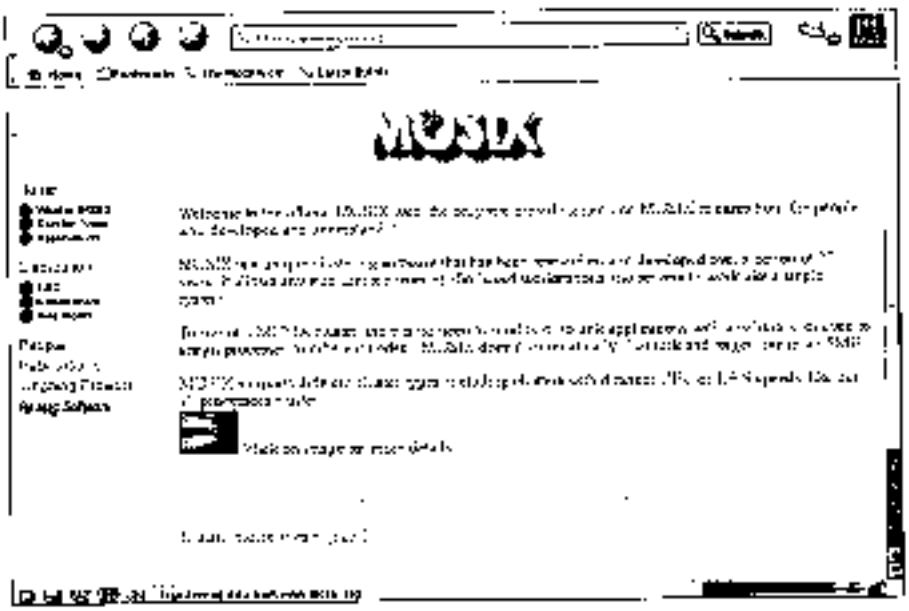


Figura 1. Mosix es un proyecto que está en constante desarrollo. Pueden encontrar más información sobre él en ***www.mosix.org****.*

Instalación de Mosix

La instalación de Mosix es relativamente sencilla en algunas distribuciones como Red Hat. Esto se debe a que los desarrolladores usan esta distribución para su desarrollo. La cosa es sencilla también en otras distribuciones como Mandrake, Debian y SuSE Linux. Todo lo que hay que hacer es descargar el archivo TGZ desde el sitio de Mosix (**www.mosix.org**), descomprimirlo y ejecutar el script de instalación **mosix.install**. Este script les hará algunas preguntas sobre el sistema, y aplicará el parche correspondiente en el núcleo. Si es necesario, lo recopilará y lo instalará automáticamente, por lo que se trata de un proceso que puede llevar un par de horas.

Para instalarlo en los demás nodos del cluster, pueden seguir los mismos pasos, o utilizar algunas herramientas que simplificarán esta tarea automatizándola. Las recomendadas por la gente de Mosix son LTSP (**www.ltsp.org**) y ClusterNFS (**clusternfs.sourceforge.net**). Si quieren obtener más información sobre el proceso de instalación, en el archivo **README** del paquete encontrarán varias formas de realizar esta tarea, como también algunos enlaces que serán de gran utilidad.

Cabe destacar que la última versión de Mosix corresponde a la versión 2.4.13 del kernel Linux. Sólo funciona con esa versión del núcleo, por lo que intentar instalarlo en un sistema con otra versión (por más que sea de la misma serie) será en vano. El equipo de Mosix ofrece también una versión de este producto para los que usan núcleos de la serie 2.2. En el sitio oficial también encontrarán RPMs para instalarlo en Mandrake (mucho más fácil), y una versión que usa ClusterNFS y no necesita disco rígido.

Configuración

Al ser un sistema que funciona de forma automática, la configuración que requiere es prácticamente mínima. Generalmente, sólo hay que editar el archivo **/etc/mosix.map**, en donde se definen, en forma de tablas, el número de nodo y su IP. En la ayuda, aparece un ejemplo muy fácil de entender: si tuviéramos un cluster de seis nodos, con tres nodos en la red 200.200.120.xxx, un nodo en 200.200.130.xxx y dos nodos en 200.200.140.xxx, la configuración sería la siguiente:

```
1 200.200.120.1 3
4 200.200.130.1 1
5 200.200.140.1 2
```

El primer número indica el nodo principal, y es, obviamente, 1. Como el primer nodo principal posee tres nodos, el segundo principal será el 4, y así sucesivamente. Luego se especifica la dirección IP de cada nodo principal y la cantidad de nodos que hay bajo la red de esa dirección.

Utilidades extra

Como se imaginarán, una supercomputadora necesitará de un administrador capaz de manejar esa inmensa cantidad de recursos de que dispone. Como siempre, los administradores necesitarán herramientas para poder llevar a cabo este tipo de tareas. Mosix incluye una serie de utilidades extra como **setpe** (para la configuración del nodo), **mon** (un monitor que nos muestra el rendimiento general del cluster, **mosctl** (para administrar el cluster), **migrate** (para migrar un proceso) y muchas más. Todas poseen su página de manual y serán de gran ayuda para el administrador.

En resumen...

Los clusters son una excelente idea cuando se necesita un sistema de supercomputación pero no se tienen los recursos económicos suficientes para invertir en él. Obviamente, esta sección del libro sólo ha intentado introducir el concepto, y es recomendable que el lector recurra a más información en la página oficial de Mosix.

Documentos oficiales

En esta sección del libro publicamos dos documentos oficiales del mundo del Software Libre. El primero de ellos es la Licencia Pública General, que explica detalladamente cómo un programa de Software Libre puede ser distribuido y modificado. El segundo documento es "Abogacía por Linux", traducido por el equipo de INSFLUG, y es una herramienta indispensable para todo aquel que quiera saber cómo "defender" el sistema operativo frente a usuarios de otros sistemas, los medios de comunicación y otros grupos no relacionados con el Software Libre. Que los disfruten.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@tectimes.com

APÉNDICE A

La Licencia Pública General

Ésta es la conocida GNU Public License (GPL), versión 2 (de junio de 1991), que cubre la mayor parte del software de la Free Software Foundation, y muchos más programas. La traducción al español ha sido revisada por Richard Stallman, pero no tiene ningún valor legal, ni ha sido comprobada de acuerdo con la legislación de ningún país en particular.
Los autores de esta traducción son:
- Jesús González Barahona.
- Pedro de las Heras Quirós.

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, EE.UU.
Se permite la copia y la distribución de copias literales de este documento, pero no, su modificación.

Preámbulo

Las licencias que cubren la mayor parte del software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, también puede aplicarla a sus propios programas.
Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio. Nuestras Licencias Públicas Generales están diseñadas para asegurarnos de que tenga la libertad de distribuir copias de software libre (y cobrar por ese servicio, si quiere), de que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer todas estas cosas.
Para proteger sus derechos, necesitamos algunas restricciones que prohiban a cualquiera negarle a usted estos derechos o pedirle que renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que le afectan si distribuye copias del software, o si lo modifica.

Por ejemplo, si distribuye copias de uno de estos programas, sea gratuitamente o a cambio de una contraprestación, debe dar a los receptores todos los derechos que tiene. Debe asegurarse de que ellos también reciben, o pueden conseguir, el código fuente. Y debe mostrarles estas condiciones de forma que conozcan sus derechos.
Protegemos sus derechos con la combinación de dos medidas:

1. Ponemos el software bajo copyright, y
2. le ofrecemos esta licencia, que le da permiso legal para copiar, distribuir y/o modificar el software.

También, para la protección de cada autor y la nuestra propia, queremos asegurarnos de que todo el mundo comprenda que no se proporciona ninguna garantía para este software libre. Si el software es modificado por cualquiera y éste, a su vez, lo distribuye, queremos que sus receptores sepan que lo que tienen no es el original, de forma que cualquier problema introducido por otros no afecte a la reputación de los autores originales.
Por último, cualquier programa libre está constantemente amenazado por patentes sobre el software. Queremos evitar el peligro de que los redistribuidores de un programa libre obtengan patentes por su cuenta, convirtiendo de facto el programa en propietario. Para evitar esto, hemos dejado claro que cualquier patente debe ser pedida para el uso libre de cualquiera, o no ser pedida.
Los términos exactos y las condiciones para la copia, distribución y modificación se exponen a continuación.

Términos y condiciones para la copia, distribución y modificación

1. Esta Licencia se aplica a cualquier programa u otro tipo de trabajo que contenga una nota colocada por el tenedor del copyright que indique que puede ser distribuido bajo los términos de esta Licencia Pública General. En adelante, «Programa» se referirá a cualquier programa o trabajo que cumpla esa condición, y «trabajo basado en el Programa» se referirá bien al Programa o a cualquier trabajo derivado de él según la ley de copyright. Esto es, un trabajo que contenga el programa o una porción de él, bien en forma literal o con modificaciones y/o traducido en otro lenguaje. Por lo tanto, la traducción está incluida sin limitaciones en el término «modificación». Cada concesionario (licenciatario) será denominado «usted».
   Cualquier otra actividad que no sea la copia, la distribución o la modificación no está cubierta por esta Licencia, está fuera de su ámbito. El acto de ejecutar el Programa no está restringido, y los resultados del Programa están cubiertos únicamente si sus contenidos constituyen un trabajo basado en el Programa, independientemente de haberlo producido mediante la ejecución del programa. Que esto se cumpla depende de lo que haga el programa.

2. Usted puede copiar y distribuir copias literales del código fuente del Programa, según lo ha recibido, en cualquier medio, supuesto que de forma adecuada y bien visible publique en cada copia un anuncio de copyright adecuado y un repudio de garantía, mantenga intactos todos los anuncios que se refieran a esta Licencia y a la ausencia de garantía, y proporcione a cualquier otro receptor del programa una copia de esta Licencia junto con el Programa.
   Puede cobrar un precio por el acto físico de transferir una copia, y puede, según su libre albedrío, ofrecer garantía a cambio de unos honorarios.

3. Puede modificar su copia o copias del Programa o de cualquier porción de él, formando de esta manera un trabajo basado en el Programa, y copiar y distribuir esa modificación o trabajo bajo los términos del apartado 1, antedicho, supuesto que además cumpla las siguientes condiciones:
   a) Debe hacer que los ficheros modificados lleven anuncios prominentes que indiquen que los ha cambiado, y la fecha de cualquier cambio.
   b) Debe hacer que cualquier trabajo que distribuya o publique, y que en todo o en parte contenga o sea derivado del Programa o de cualquier parte de él, sea licenciado como un todo, sin carga alguna, a todas las terceras partes y bajo los términos de esta Licencia.
   c) Si el programa modificado lee normalmente órdenes interactivamente cuando es ejecutado, debe hacer que, cuando comience su ejecución para ese uso interactivo de la forma más habitual, muestre o escriba un mensaje que incluya un anuncio de copyright y un anuncio de que no se ofrece ninguna garantía (o, por el contrario, que sí se ofrece garantía), y que los usuarios pueden redistribuir el programa bajo estas condiciones, e indique al usuario cómo ver una copia de esta licencia. (Excepción: si el propio programa es interactivo pero normalmente no muestra ese anuncio, no se requiere que su trabajo basado en el Programa muestre ningún anuncio).

   Estos requisitos se aplican al trabajo modificado como un todo. Si partes identificables de ese trabajo no son derivadas del Programa, y pueden, razonablemente, ser consideradas trabajos independientes y separados por ellos mismos, entonces esta Licencia y sus términos no se aplican a esas partes cuando sean distribuidas como trabajos separados. Pero cuando distribuya esas mismas secciones como partes de un todo que es un trabajo basado en el Programa, la distribución del todo debe ser según los términos de esta licencia, cuyos permisos para otros licenciatarios se extienden al todo completo y, por lo tanto, a todas y cada una de sus partes, con independencia de quién la escribió.
   Por lo tanto, no es la intención de este apartado reclamar derechos o desafiar sus derechos sobre trabajos escritos totalmente por usted mismo. El intento es ejercer el derecho a controlar la distribución de trabajos derivados o colectivos basados en el Programa.
   Además, el simple hecho de reunir un trabajo no basado en el Programa con el

Programa (o con un trabajo basado en el Programa) en un volumen de almacenamiento o en un medio de distribución no hace que dicho trabajo entre dentro del ámbito cubierto por esta Licencia.

4. Puede copiar y distribuir el Programa (o un trabajo basado en él, según se especifica en el apartado 2), como código objeto o en formato ejecutable según los términos de los apartados 1 y 2, supuesto que además cumpla una de las siguientes condiciones:
   a) Acompañarlo con el código fuente completo correspondiente, en formato electrónico, que debe ser distribuido según se especifica en los apartados 1 y 2 de esta Licencia, en un medio habitualmente utilizado para el intercambio de programas, o
   b) acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa en formato electrónico del código fuente correspondiente, a un coste no mayor que el de realizar físicamente la distribución del fuente, que será distribuido bajo las condiciones descritas en los apartados 1 y 2 anteriores, en un medio habitualmente utilizado para el intercambio de programas, o
   c) acompañarlo con la información que recibió ofreciendo distribuir el código fuente correspondiente. (Esta opción se permite sólo para distribución no comercial, y sólo si usted recibió el programa como código objeto o en formato ejecutable con tal oferta, de acuerdo con el apartado b anterior).

   Por código fuente de un trabajo se entiende la forma preferida del trabajo cuando se le hacen modificaciones. Para un trabajo ejecutable, se entiende por código fuente completo todo el código fuente para todos los módulos que contiene, más cualquier fichero asociado de definición de interfaces, más los guiones utilizados para controlar la compilación e instalación del ejecutable. Como excepción especial, el código fuente distribuido no necesita incluir nada que sea distribuido normalmente (bien como fuente, bien en forma binaria) con los componentes principales (compilador, kernel y similares) del sistema operativo en el cual funciona el ejecutable, a no ser que el propio componente acompañe al ejecutable.

   Si la distribución del ejecutable o del código objeto se hace mediante la oferta de acceso para copiarlo de un cierto lugar, entonces se considera la oferta de acceso para copiar el código fuente del mismo lugar como distribución del código fuente, incluso aunque terceras partes no estén forzadas a copiar el fuente junto con el código objeto.

5. No puede copiar, modificar, sublicenciar o distribuir el Programa, excepto como prevé expresamente esta Licencia. Cualquier intento de copiar, modificar, sublicenciar o distribuir el Programa de otra forma es inválida, y hará que cesen automáticamente los derechos que le proporciona esta Licencia. En cualquier caso, las partes que hayan recibido copias o derechos de usted bajo esta Licencia no resarán en sus derechos mientras esas partes continúen cumpliéndola.

6. No está obligado a aceptar esta licencia, ya que no la ha firmado. Sin embargo, no hay nada más que le proporcione permiso para modificar o distribuir el Programa o sus trabajos derivados. Estas acciones están prohibidas por la ley si no acepta esta Licencia. Por lo tanto, si modifica o distribuye el Programa (o cualquier trabajo basado en el Programa), está indicando que acepta esta Licencia para poder hacerlo, y todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.

7. Cada vez que redistribuya el Programa (o cualquier trabajo basado en el Programa), el receptor recibirá automáticamente una licencia del licenciatario original para copiar, distribuir o modificar el Programa, de forma sujeta a estos términos y condiciones. No puede imponer al receptor ninguna restricción más sobre el ejercicio de los derechos aquí garantizados. No es usted responsable de hacer cumplir esta licencia por terceras partes.

8. Si como consecuencia de una resolución judicial o de una alegación de infracción de patente o por cualquier otra razón (no limitada a asuntos relacionados con patentes) se le imponen condiciones (ya sea por mandato judicial, por acuerdo o por cualquier otra causa) que contradigan las condiciones de esta Licencia, ello no le exime de cumplir las condiciones de esta Licencia. Si no puede realizar distribuciones de forma que se satisfagan simultáneamente sus obligaciones bajo esta licencia y cualquier otra obligación pertinente, entonces, como consecuencia, no puede distribuir el Programa de ninguna forma. Por ejemplo, si una patente no permite la redistribución libre de derechos de autor del Programa por parte de todos aquellos que reciban copias directa o indirectamente a través de usted, entonces la única forma en que podría satisfacer tanto esa condición como esta Licencia sería evitar completamente la distribución del Programa.
   Si cualquier porción de este apartado se considera inválida o imposible de cumplir bajo cualquier circunstancia particular, ha de cumplirse el resto, y la sección por entero ha de cumplirse en cualquier otra circunstancia.
   No es el propósito de este apartado inducirle a infringir ninguna reivindicación de patente ni de ningún otro derecho de propiedad o impugnar la validez de ninguna de dichas reivindicaciones. Este apartado tiene el único propósito de proteger la integridad del sistema de distribución de software libre, que se realiza mediante prácticas de licencia pública. Mucha gente ha hecho contribuciones generosas a la gran variedad de software distribuido mediante ese sistema con la confianza de que el sistema se aplicará consistentemente. Será el autor/donante quien decida si quiere distribuir software mediante cualquier otro sistema, y una licencia no puede imponer esa elección.
   Este apartado pretende dejar completamente claro lo que se cree que es una consecuencia del resto de esta Licencia.

9. Si la distribución y/o uso del Programa está restringida en ciertos países, bien por patentes o por interfaces bajo copyright, el tenedor del copyright que coloca este Programa bajo esta Licencia puede añadir una limitación explícita de distribución geográfica excluyendo esos países, de forma que la distribución se permita sólo en o entre los países no excluidos de esta manera. En ese caso, esta Licencia incorporará la limitación como si estuviese escrita en el cuerpo de esta Licencia.

10. La Free Software Foundation puede publicar versiones revisadas y/o nuevas de la Licencia Pública General de tiempo en tiempo. Dichas nuevas versiones serán similares en espíritu a la presente versión, pero pueden ser diferentes en detalles para considerar nuevos problemas o situaciones.
Cada versión recibe un número de versión que la distingue de otras. Si el Programa especifica un número de versión de esta Licencia que se refiere a ella y a «cualquier versión posterior», tiene la opción de seguir los términos y condiciones, bien de esa versión, bien de cualquier versión posterior publicada por la Free Software Foundation. Si el Programa no especifica un número de versión de esta Licencia, puede escoger cualquier versión publicada por la Free Software Foundation.

11. Si quiere incorporar partes del Programa en otros programas libres cuyas condiciones de distribución son diferentes, escriba al autor para pedirle permiso. Si el software tiene copyright de la Free Software Foundation, escriba a la Free Software Foundation: algunas veces hacemos excepciones en estos casos. Nuestra decisión estará guiada por el doble objetivo de preservar la libertad de todos los derivados de nuestro software libre, y promover que se comparta y reutilice el software en general.

Ausencia de garantía

12. Como el programa se licencia libre de cargas, no se ofrece ninguna garantía sobre el programa, en toda la extensión permitida por la legislación aplicable. Excepto cuando se indique de otra forma por escrito, los tenedores del copyright y/u otras partes proporcionan el programa «tal cual», sin garantía de ninguna clase, bien expresa o implícita, con inclusión, pero sin limitación a las garantías mercantiles implícitas o a la conveniencia para un propósito particular. Cualquier riesgo referente a la calidad y a las prestaciones del programa es asumido por usted. Si se probase que el Programa es defectuoso, asume el coste de cualquier servicio, reparación o corrección.

13. En ningún caso, salvo que lo requiera la legislación aplicable o haya sido acordado por escrito, ningún tenedor del copyright ni ninguna otra parte que modifique

y/o redistribuya el Programa según se permite en esta Licencia será responsable ante usted por daños, incluido cualquier daño general, especial, incidental o resultante producido por el uso o la imposibilidad de uso del Programa (con inclusión, pero sin limitación a la pérdida de datos o a la generación incorrecta de datos o a pérdidas sufridas por usted o por terceras partes, o a un fallo del Programa al funcionar en combinación con cualquier otro programa), incluso si dicho tenedor u otra parte ha sido advertido de la posibilidad de dichos daños.

Cómo aplicar estos términos a sus nuevos programas

Si usted desarrolla un nuevo Programa y quiere que sea del mayor uso posible para el público en general, la mejor forma de conseguirlo es convirtiéndolo en software libre que cualquiera pueda redistribuir y cambiar bajo estos términos.
Para hacerlo, añada los siguientes anuncios al programa. Lo más seguro es añadirlos al principio de cada fichero fuente para transmitir lo más efectivamente posible la ausencia de garantía. Además, cada fichero debería tener, al menos la línea de «copyright» y un indicador de dónde puede encontrarse el anuncio completo.
<una línea para indicar el nombre del programa y una rápida idea de qué hace.>
Copyright (C) 20aa <nombre del autor>
Este programa es software libre. Puede redistribuirlo y/o modificarlo bajo los términos de la Licencia Pública General de GNU según es publicada por la Free Software Foundation, bien de la versión 2 de dicha Licencia o bien (según su elección) de cualquier versión posterior.
Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA, incluso sin la garantía MERCANTIL implícita o sin garantizar la CONVENIENCIA PARA UN PROPÓSITO PARTICULAR. Véase la Licencia Pública General de GNU para conocer más detalles.
Debería haber recibido una copia de la Licencia Pública General junto con este programa. Si no ha sido así, escriba a la Free Software Foundation, Inc., en 675 Mass Ave, Cambridge, MA 02139, EE.UU.
Añada también información sobre cómo contactarse con usted mediante correo electrónico y postal.
Si el programa es interactivo, haga que muestre un pequeño anuncio como el siguiente, cuando comienza a funcionar en modo interactivo:
Gnomovision versión 69, Copyright (C) 20aa nombre del autor.
Gnomovision no ofrece ABSOLUTAMENTE NINGUNA GARANTÍA. Para obtener más detalles, escriba «show w».
Los comandos hipotéticos «show w» y «show c» deberían mostrar las partes adecuadas de la Licencia Pública General. Por supuesto, los comandos que use pueden

llamarse de cualquier otra manera. Podrían incluso ser pulsaciones del ratón o elementos de un menú (lo que sea apropiado para su programa).
También debería conseguir que su empleador (si trabaja como programador) o su universidad (si es el caso) firme un «renuncia de copyright» para el programa, si es necesario. A continuación se ofrece un ejemplo; altere los nombres según sea conveniente:

> Yoyodyne, Inc. mediante este documento renuncia a cualquier interés de derechos de copyright con respecto al programa Gnomovision (que hace pasadas a compiladores) escrito por Pepe Programador.
> <firma de Pepito Grillo>, 20 de diciembre de 1996.
> Pepito Grillo, Presidente de Asuntillos Varios.

Esta Licencia Pública General no permite que incluya sus programas en programas propietarios. Si su programa es una biblioteca de subrutinas, puede considerar más útil permitir el enlazado de aplicaciones propietarias con la biblioteca. Si éste es el caso, use la Licencia Pública General de GNU para Bibliotecas en lugar de esta Licencia.

APÉNDICE B

"Abogacía por Linux"

Éste es el CÓMO para la Defensa de Linux, y tiene la intención de proporcionar pautas e ideas para apoyar sus esfuerzos de defensa de Linux.
Este CÓMO fue inspirado por Jon Maddog Hall cuando respondió a una demanda de guías para defender a Linux durante las actividades del Día de la Red del 96 (NetDay96, **www.netday96.com**). Jon respondió positivamente a estas guías y observó que ellas eran la base de una lista de «normas de conducta» que podrían beneficiar a la comunidad de Linux.

- Este documento está disponible en formato HTML y en su versión original en: **www.datasync.com/~rogerspl/Advocacy-HOWTO.html**.

- Nat Makarevitch, **nat@nataa.fr.eu.org**, tradujo este documento al francés: **www.freenix.org/unix/linux/HOWTO/mini/Advocacy.html**.

- Chie Nakatani, **jeanne@mbox.kyoto-inet.or.jp**, tradujo este documento al japonés: **jf.gee.kyoto-u.ac.jp/JF/JF-ftp/html/Advocacy.tar.gz**.

- Janusz Batko, **janus@krakow.linux.org.pl**, tradujo este documento al polaco: **www.jtz.org.pl/Html/mini/Advocacy.pl.html**.

- Bruno H. Collovini, **buick@microlink.com.br**, tradujo este documento al portugués: **www.microlink.com.br/~buick/dragons/op1/minihowtos/br-advocacy.html**.

- Mauricio Rivera Pineda, **rmrivera@hotmail.com**, tradujo este documento al castellano: **www.insflug.org/documentos/Abogacia-Por-Linux-Mini-Como/**.

El autor y mantenedor del CÓMO para la Defensa de Linux es Paul L. Rogers, **Paul.L.Rogers@li.org**. Los comentarios y adiciones que se propongan serán bienvenidos.
Si necesita saber más acerca del Proyecto de Documentación de Linux o acerca de los CÓMOs de Linux, siéntase libre de contactarse con él. En el caso de la comunidad hispanohablante, diríjase al INSFLUG, **www.insflug.org/**, y contáctese con el coordinador, Joshua Drake, **poet@linuxports.com**.

Debido a varias circunstancias, no he podido dedicarle el tiempo suficiente a mantener este miniCÓMO ni para interactuar con la comunidad de Linux tanto tiempo como hubiera deseado. Pido disculpas por esto, y si usted ha intentado contactarme y he sido lento en responderle, por favor perdóneme el haber sido tan desconsiderado. A pesar de que todavia tengo muchos otros compromisos, anticipo que comenzarán a demandarme menos tiempo y me permitirán ponerme al dia en mis otras facetas de la vida. Agradezco su paciencia, y quisiera extender unos agradecimientos especiales a todos los que han sacado tiempo para sugerir adiciones y correcciones.

Información sobre derechos reservados de autor

Este CÓMO tiene derechos reservados de autor © 1996 por Paul L. Rogers. Todos los derechos reservados.
Cualquier copia literal puede reproducirse o distribuirse en cualquier medio fisico o electrónico sin permiso expreso del autor.
Las traducciones igualmente son permitidas sin el permiso expreso si éstas incluyen una nota acerca de quién lo tradujo.
Pueden agregarse citas cortas sin el consentimiento previo del autor.
Los trabajos derivados y distribuciones parciales del CÓMO de Defensa de Linux

deben acompañarse ya sea con una copia literal de este archivo o con un enlace a una copia literal. La redistribución comercial está permitida y es alentada; sin embargo, al autor le gustaría ser notificado de tales distribuciones.
En resumen, deseamos promover la diseminación de esta información a través del mayor número posible de canales. Sin embargo, deseamos retener los derechos de autor de los documentos de CÓMO, y nos gustaría ser notificados de cualquier plan de redistribuir los CÓMOs.
Más aún, queremos que toda la información provista en los CÓMOs sea diseminada. Si usted tiene preguntas, por favor contáctese con el coordinador de los CÓMOs (en inglés), Joshua Drake, **poet@linuxports.com**.
Para temas relacionados con documentación en castellano, diríjase a **www.insflug.org/colaboracion**.

Introducción

La comunidad de Linux ha sabido por algún tiempo que para muchas aplicaciones, Linux es un producto estable, fiable y robusto (aun cuando no perfecto). Desafortunadamente, hay todavía mucha gente, incluidos tomadores de decisiones clave, que no son conscientes de la existencia de Linux ni de sus capacidades.
Si Linux y los muchos otros componentes que conforman una distribución de Linux han de lograr su máximo potencial, es crítico que lleguemos a los usuarios posibles y defendamos (siendo muy cuidadosos de no prometer demasiado) el uso de Linux para las aplicaciones apropiadas.
La razón por la cual muchos productos comerciales han tenido éxito en el mercado no es tanto por la superioridad técnica como por las habilidades publicitarias de la compañía. Si usted goza con el uso de Linux y le gustaría contribuir de alguna manera con la comunidad de Linux, por favor considere la posibilidad de actuar de acuerdo con alguna o algunas de las ideas expresadas en este CÓMO, y así, ayude a otros a aprender más acerca de Linux.

Información relacionada

Lars Wirzenius, antiguo moderador de **comp.os.linux.announce** y activista de Linux durante mucho tiempo, tiene también algunos pensamientos (**www.iki.fi/liw/texts/advocating-linux.html**) acerca de cómo promover el uso de Linux.

Eric S. Raymond ha escrito un análisis, disponible en **www.tuxedo.org/~esr/writings/cathedral-bazaar/index.html**, acerca de por qué el modelo de desarrollo usado por la comunidad de Linux ha tenido tanto éxito.

La comunidad de software libre ha reconocido que los términos «software libre» y «software disponible libremente» no son apropiados en todos los contextos. Para obtener más información acerca del uso del término «software de fuentes abiertas» cuando promueva el «software libre», por favor visite el servidor del movimiento Open Source, **www.opensource.org/**.

Si necesita pulir sus técnicas de venta de Linux, eche un vistazo al ensayo Linuxmanship, **http://electriclichen.com/people/dmarti/linuxmanship.html**, escrito por Donald B. Marti, Jr.

Por otra parte, en Linux PR (**www.cse.unsw.edu.au/~conradp/linux/pr/**) se discute la importancia de los comunicados de prensa hacia la comunidad de Linux. Otra forma de obtener valiosa experiencia en esta área es la de organizar un Dia de la Red (NetDay) en alguna escuela local usando las ideas presentadas en la siguiente guia: **www.netday.org/how-to/**.

El propósito de Linux International (**www.li.org**) es promover el desarrollo y el uso de Linux.

El Linux Documentation Project (Proyecto para la Documentación de Linux), en **sunsite.unc.edu/mdw/linux.html**, es un recurso de valor incalculable para los promotores de Linux.

En el Linux Center Project (**www.portalux.com/**) tiene un indice temático de recursos acerca de Linux y software libre.

El servidor de Linux Business Applications (**www.m-tech.ab.ca/linux-biz/**) ofrece un foro para que organizaciones que dependen de Linux para operaciones diarias de negocios compartan sus experiencias.

En Linux Enterprise Computing (**linas.org/linux/**) y Freely Redistributable Software in Business (**www.cyber.com.au/misc/frsbiz/**) se cubren recursos y tópicos de interés para aquellos que utilicen Linux en un ambiente comercial de empresa privada.

El propósito de Linux Advocacy Project (Proyecto para la Defensa de Linux), en **www.10mb.com/linux/**, es el de alentar a los desarrolladores de aplicaciones comerciales a ofrecer versiones nativas en Linux de su software.

El programa de Linux CD and Support Giveaway (**emile.math.ucsb.edu:8000/giveaway.html**) está ayudando a que Linux esté disponible en mayor número, por medio de la acción de alentar la reutilización de los CD-ROMs de Linux.

El servidor de Specialized Systems Consultants, Inc. (SSC), en **www.ssc.com/**, aloja al servidor de Linux Resources, **www.linuxresources.com/**, y publica la revista *Linux Journal* (**http://www.ssc.com/lj/**).

La lista de correo linux-biz (**www.lege.com/linux-biz.html**) es un foro creado para discutir sobre el uso de Linux en el ambiente empresarial.

La encuesta sobre Linux en Sistemas de Misión Crítica (**www.pckassa.com/mc-doc/**) documenta implantaciones con éxito de sistemas existentes que tienen una gran carga y están corriendo durante las 24 horas.

Existen algunas publicaciones en linea que se dedican a cubrir Linux. Éstas incluyen:

- LinuxFocus: (disponible en multitud de idiomas): **http://mercury.chem.pitt.edu/~angel/LinuxFocus/**.
- Linux Gazette: **www.ssc.com/lg/**.
- PLUTO Journal: **www.pluto.linux.it/journal/**.

Enlaces adicionales a publicaciones en linea pueden encontrarse en el Linux Documentation Project (**http://sunsite.unc.edu/LDP/links.html#misc**) y en el Linux Center Project, **www.linux-center.org/fr/informations/journals/index.html**.

Promoción del uso de Linux

- Comparta sus experiencias personales (buenas o malas) con Linux. Todo el mundo sabe que el software tiene errores y limitaciones, y si sólo tuviéramos comentarios resplandecientes acerca de Linux, no estaríamos siendo honestos. A mí me encanta comentarle a la gente acerca de mis cuatro reinicializadas (de tres programadas) del sistema en tres años.

- Si alguien tiene un problema que Linux puede resolver, ofrézcase para proporcionarle referencias a la información apropiada (páginas web, artículos de revistas, libros, consultores, etc.). Si no ha llevado a cabo la solución propuesta, exprésese en tal sentido.

- Si se encuentra disponible para hacer presentaciones acerca de Linux, regístrese en **www.ssc.com/linux/lsb.html**.

- Ofrezca su ayuda a quien esté comenzando a usar Linux. Hágale un seguimiento para asegurarse que pudo usar el sistema de manera efectiva.

- Algunas personas todavia creen que Linux y sistemas similares operan únicamente en modo texto. Hágalos conscientes de la disponibilidad de aplicaciones gráficas como Gimp (**www.gimp.org/**).

- Trate de responder a algún novato de las listas de noticias. Busque las preguntas difíciles; puede ser el único que responda, y aprenderá mucho en el proceso. Sin embargo, si no se tiene la confianza de poder responder con la solución correcta, busque a alguien más que sí pueda.

- Busque pequeñas firmas desarrolladoras de software y ofrézcales realizar una presentación acerca de Linux.

- Si la oportunidad se presenta, haga una presentación al grupo de informática de su empleador.

- Si necesita una aplicación que no esté soportada por Linux, contáctese con el vendedor y pídale una versión nativa de Linux.

- Participe en eventos de la comunidad, como el NetDay96 (**www.netday96.com**). Manteniendo como prioridad la de contribuir con el éxito del evento, use la oportunidad para permitir que otros conozcan lo que Linux puede hacer por ellos.

- Considere siempre los puntos de vista de la persona a quien le esté «vendiendo» Linux. Soporte, fiabilidad, interoperabilidad y costo son factores que un tomador de decisiones debe tener en cuenta. De los anteriores puntos, el coste es generalmente la parte menos importante de la ecuación.

- La disponibilidad de soporte es una preocupación frecuentemente mencionada, cuando se considera la adopción de Linux. Las siguientes compañías ofrecen soporte para algunos o todos los componentes de una distribución típica de Linux:

  - Caldera (**www.caldera.com/**)
    Cygnus Solutions (**www.cygnus.com/**)
  - Red Hat (**www.redhat.com/**)
  - S.u.S.E (**www.suse.com/**)

Adicionalmente, el CÓMO Linux Consultants HOWTO (**http://sunsite.unc.edu/LDP/HOWTO/Consultants-HOWTO.html**) incluye un listado de compañías proveedoras de soporte relacionado con Linux. Por supuesto, algunos de los mejores soportes se encuentran en el grupo de noticias news:**comp.os.linux** y en los grupos de noticias que contienen la palabra "linux".

- Haga notar que la producción de software de código abierto tiene lugar en un ambiente de colaboración abierta entre ingenieros de sistemas, programadores, escritores, probadores alfa y beta, y usuarios finales, lo cual frecuentemente da como resultado productos robustos y bien documentados, tales como Apache (**www.apache.org/**), GNU Emacs (**www.gnu.org/**), Perl (**www.perl.com/**) o el núcleo de Linux (**www.linuxhq.com/**).

- ¡Póngase de pie y déjese contar! Registrese en el Contador de Linux: **http://counter.li.org/**.

- Comunique los esfuerzos exitosos en la promoción de Linux a Linux International (**li@li.org**) y organizaciones similares.

- Busque un nuevo hogar para los CD-ROMs y libros de Linux que ya no necesite. Dónelos a alguien interesado en Linux, a una biblioteca pública o a un club escolar de computación. Un libro y su CD-ROM serían muy apropiados para una biblioteca. En todo caso, por favor asegúrese de no violar derechos de autor al hacer público el CD-ROM. Informe también a las directivas de la biblioteca que el material en CD-ROM es distribuible de forma gratuita. Haga un seguimiento para asegurarse de que quede disponible en los armarios.

- Cuando compre libros acerca de programas distribuidos con Linux, deles preferencia a los libros escritos por el autor del programa. Los beneficios que los autores reciben de las ventas de los libros pueden ser la única retribución económica que ellos obtengan a cambio de sus esfuerzos.

- Aliente a los sitios basados en Linux a que registren su sitio en la página de Powered by Linux (**http://sunsite.unc.edu/LDP/powered.html**) y sugiérales que presenten en sus sitios avisos relativos a Linux (**www.cse.unsw.edu.au/~conradp/banners/**), Apache (**www.apache.org/**), GNU (**www.gnu.org/**), Perl (**www.perl.com/**), etc.

- ¡Participe! Si usted se ha beneficiado del software de fuentes abiertas (**www.opensource.org/**), por favor, considere la posibilidad de ayudar a la comunidad del software libre por medio de algunas de estas acciones:
  - Informando detalladamente de fallos en el programa.
  - Escribiendo documentación.
  - Creando trabajo artístico.
  - Suministrando asesoría como administrador.
  - Sugiriendo mejoras.
  - Proporcionando soporte técnico.
  - Contribuyendo con programas.
  - Donando equipo.
  - Dando apoyo financiero.

Normas de conducta

- Como representante de la comunidad de Linux, participe en las listas de distribución de correo y en las discusiones de los grupos de noticias con actitud profesional. Huya del uso de lenguaje vulgar y las ofensas personales. Considérese a sí mismo como

miembro virtual de la corporación, teniendo al Sr. Torvalds como director de ésta. Sus palabras podrán tanto enaltecer como degradar la imagen que el lector se forme de la comunidad Linux.

- Evite las exageraciones y las pretensiones sin fundamento a toda costa. No es profesional y generará discusiones improductivas.

- Una respuesta meditada y bien razonada no sólo proveerá de conocimiento profundo a los lectores; también incrementará el respeto por su conocimiento y sus habilidades.

- No muerda el anzuelo si lo incitan a la guerra verbal. Muchas veces esto degenera en argumentos del tipo «Mi sistema operativo es mejor que el suyo». Describamos en forma precisa las capacidades de Linux y dejémoslo ahí.

- Recuerde siempre que si insulta o le falta el respeto a alguien, su experiencia negativa puede llegar a ser compartida por muchos otros. Si llega a ofender a alguien, por favor trate de reparar su acción.

- Céntrese en lo que Linux tiene para ofrecer. No hay necesidad de vejar a la competencia. Contamos con un producto bueno y sólido que se sostiene por sí mismo.

- Respete el uso de otros sistemas operativos. Es cierto que Linux es una plataforma maravillosa, pero no satisface necesariamente las necesidades de todo el mundo.

- Refiérase a cualquier otro producto por su propio nombre. No se gana nada intentando ridiculizar una compañía o sus productos usando «ortografías creativas». Si esperamos respeto por Linux, entonces debemos respetar los demás productos.

- Dé crédito a quien es debido. Linux es tan sólo el núcleo. Sin los esfuerzos del proyecto GNU, MIT, Berkeley y muchos otros muy numerosos para ser mencionados, el núcleo de Linux no sería útil para la mayoría.

- No insista en que Linux es la única respuesta para una aplicación particular. Así como la comunidad de Linux venera la libertad que Linux les provee, las soluciones únicas basadas en Linux coartaría a otros su libertad. Habrá casos en los que Linux no sea la respuesta. Sea el primero en reconocerlo y ofrezca otra solución.

Grupos de usuarios

- Participe en grupos locales de usuarios. Hay un índice de Grupos de Usuarios de Linux registrados en **http://sunsite.unc.edu/LDP/linux.html#general**, que es parte del LDP, **http://sunsite.unc.edu/LDP/**. Si no existe en su región, inicie uno.

- El CÓMO User Group HOWTO (**http://sunsite.unc.edu/LDP/HOWTO/User-Group-HOWTO.html**) cubre muchos aspectos relativos a la creación de un grupo de usuarios y discute sobre la importancia de la abogacía por Linux como uno de los principales objetivos de un grupo de usuarios.

- Suministre conferenciantes a las organizaciones interesadas en Linux.

- Publique comunicados de prensa (**www.cse.unsw.edu.au/~conradp/linux/pr/**) acerca de sus actividades para los medios de comunicación locales.

- Ofrézcase como voluntario para configurar un sistema Linux que satisfaga las necesidades de alguna organización local comunitaria. Por supuesto, el proceso de instalación debe incluir el entrenamiento para que el usuario de la comunidad pueda utilizar el sistema; igualmente, provéale una adecuada documentación para el mantenimiento del sistema.

- Discuta el CÓMO de la Abogacía por Linux en una reunión. Promueva discusiones y aliente la propuesta de nuevas ideas.

Relaciones con distribuidores comerciales

- Cuando contemple la compra de hardware, pregúntele al distribuidor acerca del soporte de Linux y sobre experiencias de otros usuarios con el producto en un ambiente Linux.

- Considere soportar a los distribuidores comerciales que venden productos y servicios basados en Linux. Aliéntelos a que listen sus productos en el CÓMO Linux Commercial HOWTO (**http://sunsite.unc.edu/LDP/HOWTO/Commercial-HOWTO.html**).

- Apoye a distribuidores comerciales que donen una parte de sus ingresos a organizaciones tales como la Free Software Foundation: **www.gnu.org/help/help.html**, el Linux Development Grant Fund, **www.li.org/About/Fund/Welcome.html**, el XFree86 Project, **www.xfree86.org/donations.html**, o Software in the Public Interest, **www.debian.org/donations.html**. De ser posible, realice una donación personal a estas u otras organizaciones que apoyen el software de fuentes abiertas, **www.opensource.org/**. No olvide que algunos empleadores ofrecen un programa *matching gift*.

- Si necesita una aplicación que no sea soportada en Linux, contacte al distribuidor comercial y pídale una versión nativa en Linux.

Relaciones con los medios de comunicación

- Linux International está recolectando unos recortes de prensa (**www.li.org/li/resources/pressclippings.shtml**) que mencionan a Linux, GNU y otros programas libremente distribuibles. Cuando vea un artículo con tales características, por favor envíe la siguiente información relativa al artículo, a **clippings@li.org**:
  - Nombre de la publicación.
  - Dirección donde se pueda contactar al editor.
  - Nombre del autor.
  - Dirección donde se pueda contactar al autor.
  - Título del artículo.
  - Número de la página donde comienza el artículo.
  - URL si el artículo se encuentra disponible en Internet.
  - Un resumen del artículo, incluida su opinión.

- Si usted cree que Linux no recibió un trato justo en algún artículo, revista o columna de prensa, envíe los detalles, incluida la anterior información, a **li@li.org**, para que la respuesta apropiada sea enviada al editor.
- Si está involucrado con un proyecto relativo a Linux, expida comunicados de prensa (**www.cse.unsw.edu.au/~conradp/linux/pr/**) a los servicios de noticias apropiados, de manera regular.

LA BIBLIA DE

Servicios al lector

Dentro de esta sección incluimos una serie de comandos útiles de ayuda, la clásica guía de sitios web, un glosario de términos técnicos y bibliografía recomendada.

SERVICIO DE ATENCIÓN AL LECTOR: lectores@redusers.com

Comandos útiles de ayuda

Al utilizar el sistema, los usuarios de GNU/Linux no se encuentran solos. Existen algunos comandos y parámetros que permiten obtener ayuda sobre un comando o una aplicación en especial. Lo primero que debe hacer el usuario es verificar si existe una página del manual de dicho comando. Por ejemplo, si el usuario quiere obtener ayuda sobre el comando **ls**, sólo tiene que tipear:

```
man ls
```

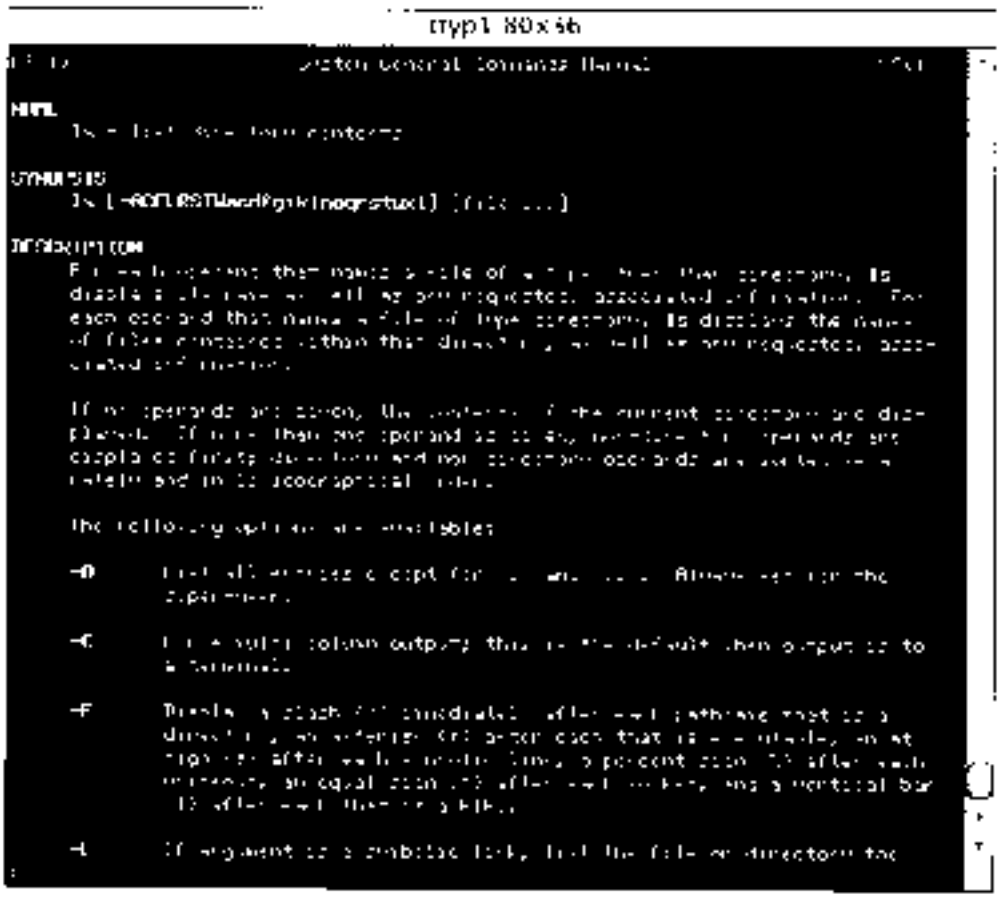


Figura 1. *Las páginas del manual son la fuente básica de información sobre comandos y aplicaciones de GNU/Linux.*

Y aparecerá el sistema de manuales con la ayuda en pantalla. Muchas veces esta ayuda estará en idioma español, otras veces no. En realidad, eso depende de que el grupo de traducción de aplicaciones y documentos de GNU/Linux haya llegado a los datos que ustedes están manejando.

Para salir del sistema de manuales, presionen la tecla **q**.

Si obtienen un mensaje de error, entonces es muy probable que dicha aplicación o

comando no incluya páginas de manual. Todavia quedan opciones. Prueben agregando el parámetro **–help** a su comando:

```
ls –help
```

Lo que obtendrán en pantalla será una escueta guía de uso de dicho comando, compuesta por un listado de parámetros y sus respectivas descripciones de acción.
Por último, recuerden que en los directorios **/usr/doc** y **/usr/share/doc** se encuentran almacenadas las documentaciones de todos los comandos y las aplicaciones que ustedes tengan instalados en su sistema. Probar con algún editor de textos puede ser una buena idea.

Los documentos CÓMO

Estos documentos intentan responder a las preguntas "¿Cómo?" formuladas por los usuarios. Generalmente, se escriben documentos relacionados con la configuración, la programación y la administración del sistema. Es muy raro ver un HOW-TO que sirva sólo para la utilización de una aplicación en particular.
Pueden encontrar los HOW-TO en idioma inglés, escritos por los miembros del Linux Documentation Proyect, en **www.linuxdoc.org**; o en castellano, traducidos por el Insflug, en **www.insflug.org**. A continuación, los CÓMOs del Insflug disponibles, con los nombres de su autor y su traductor, y la breve descripción redactada por ellos.

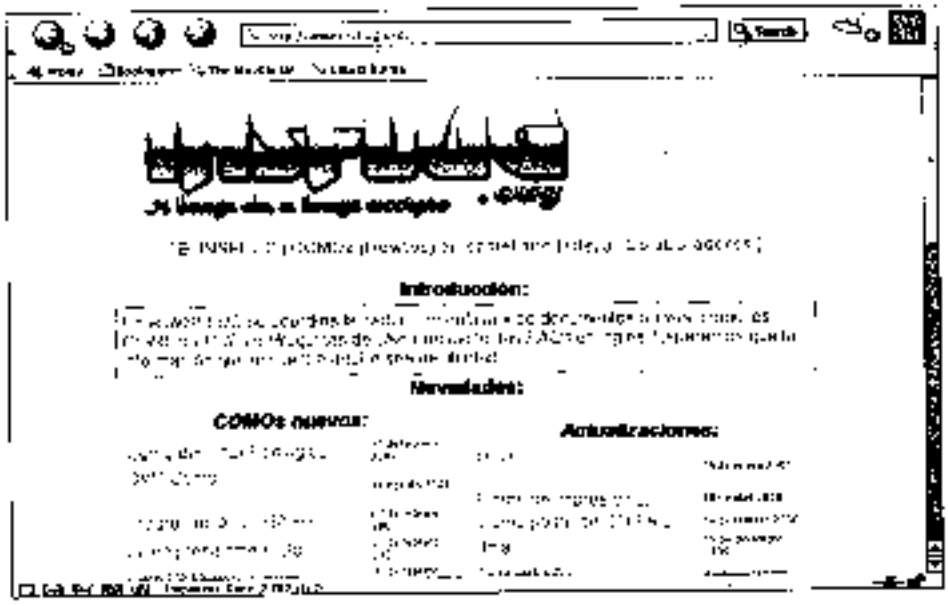


Figura 2. El sitio insflug.org es una fuente de recursos en español esencial a la hora de buscar documentación

Servidor-IRC-Como

Diego Berrueta Muñoz, berrueta@geocities.com, y **José Alfredo Oslé Posadas, aosle@pinon.ccu.uniovi.es**.
Este documento describe los pasos y el software necesarios para crear un servidor de IRC con Linux: el demonio de IRC, el bot de gestión del servidor y un programa cliente de IRC.

ifmail-COMO

Juan José Amor, jjamor@ls.fi.upm.es (fido 2:341/12.19).
Este documento pretende ser una pequeña ayuda para los usuarios de Linux pertenecientes a redes del tipo Fidonet. No puede ser perfecta, ya que viene a ser un resumen de mi propia experiencia, y tampoco tengo mucha...

PCMCIA-COMO

David Hinds, dhinds@hyper.stanford.edu.
Traducido por David Limón Romero, **dlr@cuates.pue.upaep.mx**.
Este documento describe cómo instalar y usar los servicios de las tarjetas PCMCIA con Linux. Las últimas versiones de este documento pueden encontrarse siempre en **ftp://hyper.stanford.edu/pub/pcmcia/doc**. *La versión en HTML está en* **http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html**.

Software-RAID-COMO

Jakob stergaard, jakob@ostenfeld.dk.
Traducido por Juan Piernas Cánovas, **piernas@ditec.um.es**.
Este CÓMO describe cómo usar un RAID software bajo Linux. Deberia usar los parches RAID disponibles en **ftp://ftp.fi.kernel.org/pub/linux/daemons/raid/alpha**. *El CÓMO original en inglés se puede encontrar en* **http://ostenfeld.dk/~jakob/Software-RAID-.HOWTO/**.

Mp3-COMO

Philip Kerr, phil@websentric.com.
Traducido por: Ariel Graneros, **larocka@yahoo.com**.
Este texto describe el hardware, el software y los procedimientos necesarios para comprimir y escuchar archivos MP3 bajo Linux.

Propiedad Intelectual-COMO

Juan Antonio Martínez, jantonio@dit.upm.es.
La Propiedad Intelectual se puede definir como el conjunto de derechos que corresponden a determinadas personas sobre las obras que son creación de su inteligencia.

Grabadoras-COMO

Leandro Terrés, lord_It@retemail.es, y Francisco J. Montilla, pacopepe@insflug.org.
Introducción en el uso de grabadoras de CD R, CD-RW y DVD bajo Linux, así como descripción del uso de los programas necesarios. Este documento se diferencia del **Creación-CDRoms-Como** *por su enfoque, más centrado en el uso de distintos tipos de grabadoras, y las últimas novedades de software. Es conveniente que también lean el* **Creación-CDRoms-Como,** *enfocado en la plataforma SCSI, que profundiza más en los aspectos técnicos de la tecnología CD-R y en sus posibilidades.*

Redes-En-Linux-COMO

Autor actual: **Joshua Drake (Poet), poet@linuxports.com.**
Autores originales: **Terry Dawson** (autor principal), **terry@perf.no.itg.telstra.com.au; Alessandro Rubini, rubini@linux.it** (mantenimiento).
Traducido por: **Ricardo Javier Cárdenes Medina, a1402@serdis.dis.ulpgc.es.**
Este Cómo es la base para entender la evolución de las capacidades de Linux para tratar con redes informáticas. Es el punto de partida para aprender todo sobre el mantenimiento de redes TCP/IP, la configuración de los archivos relacionados con la Red, y hay un amplio capítulo sobre configuración de dispositivos físicos. En definitiva, un documento muy exhaustivo que vale la pena leer.

Abogacía-Por-Linux-COMO

Este documento proporciona sugerencias acerca de cómo la comunidad de Linux puede abogar efectivamente por el uso de Linux.

Bash-Prompt-COMO

Giles Orr, giles@interlog.com.
Traducido por **Iosu Santurtún, iosu@bigfoot.com.**
En este documento se comenta la creación y el manejo de prompts de terminales en modo texto y X, incluidas secuencias estándar de escape que proporcionan el nombre de usuario, el directorio actual de trabajo, la hora, etc. Se hacen sugerencias más complejas sobre cómo modificar las barras de título de las terminales X, cómo usar funciones externas para proporcionar información en el prompt, y cómo usar colores ANSI.

Quakel-II-COMO

Álvaro Villalba Poncet, alvarovp@mad.servicom.es.
Guía para configurar Quake I y II *en Linux.*

Terminales-COMO

Marc Malagelada Duch, marc@hades.udg.es.
Maquetador Linuxdoc-SGML: Antonio Ismael Olea González, **olea@poboxes.com**
2:345/108.9@fidonet.org.
Todo sobre la instalación y la configuración de terminales en sistemas Linux, desde diagramas y técnicas de soldadura de los cables, hasta los detalles de configuración del hardware y del software.

Xfree86-COMO

Por **Matt Welsh**, **mdw@sunsite.unc.edu**.
Traducido por **Francisco José Montilla**, **pacopepe@insflug.org**.
Este documento describe cómo obtener, instalar y configurar la versión 3.1.1 de la variante XFree86 del Sistema XWindow (X11R6) para sistemas Linux. Es una guía paso a paso para configurar XFree86 en su sistema.

UmsDos-COMO

Jacques Gelinas, jacques@solucorp.qc.ca.
Traducido por: **Carlos García Arques**, **cgarcia@dit.upm.es**.
Umsdos es un sistema de ficheros de Linux sobre uno FAT. Ofrece una alternativa al sistema de ficheros ext2. Su objetivo principal es conseguir una coexistencia más fácil con los datos de una partición FAT, compartiéndola. En este documento primero se describe cómo usar UmsDos con diversas configuraciones, y después se describe cómo funciona y se da alguna información. Se deja para el lector la decisión de si es una buena elección para él.

Uso-Impresión-COMO

Matt Foster, **mwf@engr.uark.edu**.
Traducido por: **Ricardo Javier Cárdenes Medina**, **a1402@correo.dis.ulpgc.es**.
Este documento describe cómo usar el sistema de spooling para impresoras de líneas que provee el sistema operativo Linux. Si está buscando una guía para poner a punto su entorno de impresión, lean **Configuración de Impresion Como**.

UUCP-COMO

Vince Skahan, **vince@halcyon.com**.
Traducido por Eduardo Calatayud, **ecalatayud@st.upc.es**.
Este documento describe la instalación de UUCP bajo Linux. Necesitan leer este documento si tienen previsto acceder a sistemas remotos vía UUCP mediante módem, por conexión directa o a través de Internet.

Term-COMO

Patrick Reijnen, **patrickr@bart.nl**.
Traducción de Alfonso Belloso, **alfon@bipv02.bi.ehu.es**.
Se trata de una guía detallada de configuración del programa de comunicaciones term en Linux.

Spanish-COMO

Gonzalo García Agulló, **Gonzalo.Garcia-Agullo@jrc.es**.
Ésta es la primerísima edición del ***Linux en Castellano COMO****. La audiencia a la que va destinado este documento es la creciente familia de linuxeros españoles y latinoamericanos, por lo que este CÓMO, por supuesto, está en castellano. Algunos no hispanohablantes han solicitado información sobre Linux en entorno en español, bien sea para el desarrollo de software o su soporte en castellano, o simplemente para entrar en contacto con la comunidad linuxera hispanohablante. Sea cual sea su caso, bienvenidos...*

Sonido-COMO

Jeff Tranter, **jeff_tranter@mitel.com**.
Traducido por Sergio Alonso Manzanedo, **sa.manzanedo@vll.servicom.es**.
Este documento describe el soporte de sonido para Linux. Enumera el hardware de sonido soportado, explica cómo configurar los drivers del núcleo y contesta a las preguntas más frecuentes. Intenta enseñar más rápidamente a los nuevos usuarios y reducir el gran tráfico en los grupos de noticias de Usenet.

Serie-COMO

Greg Hankins, **greg.hankins@cc.gatech.edu**.
Traducido por Jesús Jiménez Sánchez, **jesus.jimenez@writeme.com**.
Este documento describe cómo configurar los dispositivos de comunicación serie en una máquina Linux.

Samba-COMO

David Wood, dwood@plugged.net.au.
Traducido por: Ricardo Javier Cárdenes Medina, **a1402@correo.dis.ulpgc.es.**
Este documento describe la manera de usar el paquete Samba, que dota a Linux de soporte para el protocolo Session Message Block (SMB), también llamado NetBIOS o LanManager.

RPM-COMO

Donnie Barnes, djb@redhat.com.
Traductor: Antonio Ismael Olea González, **olea@iname.com 2:345/108.9@fidonet.org.**
Este documento describe el uso del formato de paquetes de instalación que se ha convertido en estándar de facto, el RPM (RedHat Package Manager).

RDSI-COMO

Antonio Verdejo Garcia, averdejog.galileo@nexo.es, y Francisco J. Montilla, **pacopepe@insflug.org**.
Este CÓMO explica la forma de configurar tarjetas pasivas RDSI para conexiones de red PPP con Linux (a Internet, servidores...). Describe los pasos para dar soporte tanto físico como lógico, así como el método de conexión, con uno y dos canales, y con llamada bajo demanda.

Reproduccion-De-Sonido-COMO

Yoo C. Chung, http://laplace.snu.ac.kr/~wacko/_wacko@laplace.snu.ac.kr.
Traducción: Fernández Pradier Maurice, **batsman@geocities.com.**
Este documento contiene una lista de las aplicaciones para Linux capaces de reproducir diversos formatos de sonido.

Pilot-COMO

David H. Silber, pilot@orbits.com.
Traducido por Ángel López, **alogo@mx2.redestb.es.**
Este documento explica cómo usar su PalmPilot con un sistema Linux. A pesar de que los documentos CÓMO están orientados al uso del sistema operativo Linux, éste en particular no depende de la versión de UNIX utilizada.

Programacion-Serie-COMO

Peter H. Baumann, Peter.Baumann@dlr.de.
Traducción de Pedro Pablo Fábrega, **pfabrega@arrakis.es.**
Este documento describe cómo programar comunicaciones con dispositivos sobre puerto serie en una máquina Linux.

PPP-COMO

Al Longyear, longyear@netcom.com.
Traducido por Rafael Agundo (INSFLUG), **ragundo@bitmailer.net.**
Este documento contiene una lista con las preguntas más habituales (FAQ, o PUF, Preguntas de Uso Frecuente, en castellano) sobre PPP para Linux (junto con sus respuestas). No es realmente un CÓMO, pues se adapta más al formato clásico de pregunta/respuesta de las PUFs.

MetaFAQ-COMO

Michael K. Johnson, johnsonm@nigel.vnet.net.
Ésta es la Meta-FAQ para Linux. Es, principalmente, una lista de preciadas fuentes de información. Remítanse a esas fuentes si quieren aprender más sobre Linux o tienen problemas y necesitan ayuda. Lars Wirzenius (wirzeniu@cc.helsinki.fi) escribió la primera versión de este documento, y ahora está mantenido por Michael K. Johnson (johnsonm@nigel.vnet.net).

Nis-COMO

Andrea Dell'Amico, Mitchum DSouza, Erwin Embsen, Peter Eriksson.
Traducción de Carlos Martinez Txakartegi, **txakar@rigel.deusto.es.**
Configuración y detalles de NIS(YP), NIS+ y NYS (Yellow Pages), sistemas de autenticación distribuida.

IP-Masquerade-COMO

Original de Ambrose Au, ambrose@writeme.com.
Traducción de Xosé Vázquez, **xose@ctv.es.**
Este documento describe cómo activar la función IP Masquerade en un servidor Linux, para poder conectar a Internet, mediante su máquina Linux, ordenadores que no tienen registrada una dirección IP de Internet.

Linuxdoc-Ejemplo

Matt Welsh, **mdw@cs.cornell.edu**.
Traducido por Fco. José Montilla, **pacopepe@insflug.org**.
Este documento es un breve ejemplo del uso del DTD Linuxdoc-SGML.

Linuxdoc-COMO

Por **Matt Welsh**. Puesto al dia por **Greg Hankins**, **greg.hankins@cc.gatech.edu**.
Traducido por Francisco José Montilla, **pacopepe@insflug.org**.
Este documento es una guia de usuario del sistema Linuxdoc-SGML, un formateador basado en SGML que permite obtener varios tipos de formatos de salida. Se pueden producir archivos de texto plano (ASCII e ISO-8859-1), DVI, PostScript, HTML, GNU info, LyX y RTF a partir de un solo fuente SGML. Esta guia documenta el paquete Linuxdoc SGML en su versión 1.5.

Linux-Tips-COMO

Vince Reed, **reedv@rpi.edu**.
Traducido por Javier Gracia, 2:343/143.
Este documento describe ideas para ahorrar tiempo y técnicas para hacer de Linux un sistema más fácil de configurar y de usar. Todo lo incluido aqui es demasiado general como para ser incorporado en cualquiera de los otros HowTos.

Kernel-COMO

Brian Ward, **bri@blah.math.tu-graz.ac.at**.
Traducción de Juan José Amor, **jjamor@ls.fi.upm.es**.
Se trata de una guia detallada de la configuración del núcleo, que cubre detalles de compilación y de actualizaciones. El traductor ha intentado respetar el documento original, aunque no ha podido evitar añadir cosas de su propia cosecha, sobre todo en lo que respecta a opciones aún no documentadas.

Inn-Suck-Como

Pedro P. Fábrega Martínez, **pfabrega@arrakis.es**.
Este documento explica cómo configurar un servidor de news en su máquina Linux. Está enfocado en las personas que usan Linux en su casa y les gustaria tener un pequeño servidor local. También, para quien quiera dar servicio de noticias a una red local con direcciones privadas. Este documento no está dirigido a quienes quieren

ejecutar un servidor de news en una gran red, aunque la mayoría de las cosas las podría aplicar también. Por último, se supone que quien pretende instalar inn+suck tiene ya ciertos conocimientos no elementales.

Infosheet-COMO

Michael K. Johnson, johnsonm@nigel.vnet.net.
Este documento proporciona una información básica sobre el sistema operativo Linux, e incluye una explicación de Linux, una lista de características, algunos requerimientos y algunos recursos.

Accesibilidad-COMO

Michael De La Rue, access-howto@ed.ac.uk..
Traducido por Iosu Santurtún, **webmaster@sidam.es.**
El Linux Accesibilidad-COMO cubre el uso de tecnología adaptativa con Linux; en particular, la utilización de tecnología adaptativa para hacer Linux accesible a aquellos que no pueden usarlo de otra forma. También cubre áreas en las que Linux puede emplearse dentro de soluciones de tecnología adaptativa más generales.

Fuente-ISO-COMO

Urko Lusa Oiza, ulusa@frodo.com.
Este documento explica cómo usar una fuente iso-8859-1 en el modo texto de Linux, en vez de la que éste trae por defecto en todas las distribuciones que conocemos, y que no deja de ser una especie de apaño que arrastra el lastre de la página de códigos de IBM.

Ftp-Anonimo-COMO

Rodolfo García Peñas, kix@mad.servicom.es.
Este COMO describe el modo de instalar, configurar y mantener un servidor FTP de una manera fácil y rápida.

Feddi-COMO

Manuel Soriano, manu@ctv.es.
Este documento deriva del famoso feddi.como que acompaña a los paquetes del FEddi+bt, y éste está basado en la versión 0.5.

Dosemu-COMO

Editado por **Mike Deisher, deisher@dspsun.eas.asu.edu.**
Traducido por Gerardo Fernández Navarrete, **azor@freenet.hut.fi.**
Este COMO trata sobre el popular emulador de DOS para Linux, dosemu.

Dos-a-Linux-COMO

Guido Gonzato, guido@ibogfs.df.unibo.it.
Traducido por David Martin Carreño, **davefx@bigfoot.com**.
Este documento CÓMO está dedicado a todos los (¿próximamente anticuados?) usuarios de DOS que acaban de decidir pasarse a Linux, el clónico gratuito de UNIX para ordenadores x86. Dadas las similitudes entre DOS y UNIX, el propósito de este documento es ayudar al lector a traducir su conocimiento de DOS al entorno Linux, con todo lo que ello lleva de productivo.

DNS-COMO

Nicolai Langfeldt, janl@math.uio.no. Traducción: Pedro Pablo Fábrega Martinez, **pfabrega@arrakis.es** , julio 1997.
Cómo ser un administrador DNS en poco tiempo.

Cortafuegos-COMO

David Rudder, drig@execpc.com.
Traducido por Carlos Garcia Arques, **cgarcia@isabel.dit.upm.es**.
El objetivo de este documento es enseñar las bases de la instalación de un cortafuegos mediante un PC y Linux. También trata sobre la instalación y el uso de servidores Proxy, dispositivos con los que se consigue un mayor nivel de acceso a la Internet desde detrás de un cortafuegos. Título original: "Firewalling and Proxy Server HOWTO". Título alternativo propuesto por el autor: "The ride of the lonely local area network", es decir, "La balada de la red local solitaria".

CDRom-COMO

Jeff Tranter, jeff_tranter@pobox.com.
Traducido por Carlos Fernández Moro, **uov00655@correo.uniovi.es**.
Ésta es la traducción del CD-ROM-HOWTO. En este documento se explica la manera de instalar, configurar y utilizar unidades de CD-ROM bajo Linux. Incluye una lista de los modelos soportados y las respuestas a las PUFs (Preguntas de Uso Frecuente). La intención no es otra que la de capacitar a los usuarios noveles en un nivel básico y aligerar la correspondencia en los grupos de noticias de Usenet y listas de correo.

Configuración-Impresión-COMO

Grant Taylor, gtaylor@cs.tufts.edu, y **Brian McCauley, B.A.McCauley@bham.ac.uk.**
Traducción: Francisco Escarpa, **escarpa@impronta.es**.
Una guía de cómo imprimir y ver documentos bajo el sistema operativo Linux.

BitchX-COMO

Álvaro Villalba Poncet, alvarovp@mad.servicom.es.
BitchX es un cliente para el IRC, pero NO LO ES PARA LAS X; es en modo texto puro y duro. Básicamente, BitchX es el IRCII con los scripts integrados en el código fuente C, lo que lo hace más eficiente, además de dotarlo de muchas funcionalidades.

Vesabf-Mini-COMO

Autor: **Alex Buell, alex.buell@tahallah.demon.co.uk.**
Traducido por: Jesús Martínez Mateo, **jmartinez@sportec.es**.
Este documento describe cómo usar el dispositivo ***vesafb*** *bajo Linux con una tarjeta gráfica compatible VESA 2.0 en plataformas Intel.*

Instalación-Oracle-Mini-COMO

Luis M. Cruz, lcruzva@clientes.unicaja.es, y **Ángel Carrasco, karrasko@arrakis.es.**
Existen programas cuya instalación es difícil, existen programas cuya configuración es difícil, existen programas cuyo manejo es difícil y existen programas cuya instalación, configuración y manejo son difíciles, por ejemplo: ORACLE. Este Mini-Como tiene una intención especial: ayudar al usuario a realizar por sí mismo una instalación de Oracle.

GPG-Mini-COMO

Michael Fischard v. Mollard, fischer@math.uni-goettingen.de (versión alemana).
Brenno J.S.A.A.F. de Winter, brenno@dewinter.com (versión inglesa).
Horacio, homega@ciberia.es (versión en castellano).
Este documento trata sobre la instalación, la configuración y el uso de Gnu Privacy Guard (GnuPG), un sistema de codificación de código libre y desarrollo abierto, compatible con OpenPGP. Con el fin de mantener este programa totalmente libre, se ha evitado el uso de algoritmos con patentes propietarias restrictivas, como las de IDEA y RSA. El documento original fue escrito en alemán por Michael Fischard v. Mollard, y posteriormente traducido al inglés, y revisado en algunos puntos, por Brenno J.S.A.A.F. de Winter. La traducción de este documento al castellano se ha llevado a cabo a partir de la versión inglesa. El capítulo 5 se ha añadido en la versión en castellano, y también se han incluido algunos recursos y otra información. Ésta es una

revisión de la versión 0.1.2, y no incluye ninguna temática nueva, tan sólo su conversión de código HTML a código SGML para su posterior reconversión a otros formatos. También se han corregido algunos errores de forma o traducción.

Navegador-Público-Mini-COMO

Donald B. Marti Jr., **dmarti@best.com**.
Traducido por Mauricio Rivera Pineda, **rmrivera@hotmail.com.**
La idea básica es la de dar acceso WWW al público, pero limitando la posibilidad de desconfigurar el sistema, limitando su entorno únicamente al navegador.

Puntero-X-Gigante-Mini-COMO

Jörg Schneider, **joerg.schneider@ira.uka.de**.
Traducido por Iosu Santurtún, **iosu@bigfoot.com**.
Este documento describe cómo usar punteros agrandados con el sistema XWindow.

Linux-NT-Loader

Bernd Reichert, reichert@dial.eunet.ch.
Traducción: Diego Berrueta, **berrueta@geocities.com.**
Este documento describe el uso del Windows NT boot loader para arrancar Linux. Estos procedimientos han sido probados con Windows NT 4.0 WKS y Linux 2.0.

Bzip2-COMO

David Fetter, dfetter@best.com.
Traducido por: Joaquín Cuenca Abela, **jcuenca@patan.eleinf.uv.es**.
Este documento explica cómo utilizar el nuevo compresor bzip2.

Reescritura-Direcciones-En-Sendmail-COMO

Thomas Roessler, roessler@guug.de.
Traducción: David Marín Carreño, **davefx@bigfoot.com.**
Este documento es una breve descripción de cómo modificar el fichero de configuración de sendmail para un acceso telefónico doméstico.

ZIP-COMO

Grant Guenther, grant@torque.net.
Traductor: Iñaki Martínez Díez, **imd@redestb.es**; revisión y actualización: **pacopepe@insflug.org**.

La unidad Iomega ZIP es una unidad de disco extraible y de moda. Está disponible en tres versiones principales: la hay con interfaz SCSI, con IDE, y otra que se conecta a un puerto paralelo. Este documento describe cómo usar el ZIP con Linux. Se deberia leer en conjunción con el HOWTO SCSI, a menos que se posea la versión IDE.

SLIP-COMO

Manuel Soriano, m **soriano@dapsys.ch**.
Maquetador Linuxdoc-SGML: Antonio Ismael Olea González, **olea@iname.com** 2:345/108.9@fidonet.org.
Cómo configurar Linux para usar TCP/IP por cable serie mediante SLIP.

SAC-COMO

Copyright (C) 1996 **Alexander O. Yuriev**, **alex@bach.cis.temple.edu**.
Traducido por Salvador Fernández Barquín, **sferbar@internetica.net.mx**.
Este documento describe cómo compilar, instalar y configurar CFS, el Cryptographical File System.

Servidor-Raices-NFS-COMO

Ofer Maor, ofer@hadar.co.il.
Traducido por Rodolfo Pilas, **rodolfo@linux.org.uy**.
El objetivo de este Mini-COMO es explicar cómo crear directorios en un servidor que los clientes montarán como su directorio raiz por NFS.

Servidor-Intranet-COMO

Pramod Karnad, karnad@indiamail.com.
Traducido por Salvador Fernández Barquin, **sferbar@internetica.net.mx**.
Este documento describe cómo configurar una intranet usando Linux como servidor que enlace UNIX, Netware, NT y Windows juntos. Desde ahora, estableciendo la conexión con la máquina, Linux estará otorgando acceso transparente a todas las plataformas. Se proporcionan explicaciones detalladas para configurar HTTP usando el servidor NCSA y cómo conectarse a éste usando clientes TCP/IP desde Novell, Microsoft Windows 3.1, 3.11, Win95, WinNT y MacTCP para Apple PowerMac.

Lscolor-COMO

V. inglesa: **Thorbjoern Ravn Andersen**, **ravn@imada.ou.dk**.
V. española: Tomás Hidalgo, Guy Paul Gallardo, **thidalgo@alboran.ualm.es**
El comando ls incluido en la distribución Slackware de Linux permite color. (Slackware

2.0.2 y núcleo 1.1.54 de Linux. Funciona también con agetty, o bien con mgetty_psl. Puede que los colores predeterminados no les gusten. Este documento tiene como objetivo explicar el principio de funcionamiento y configuración de colores usados por este comando.

Ratones 3 botones-COMO

Geoff Short, grs100@york.ac.uk.
Traducido por Iñaki Martínez Díez, **imd@redestb.es**.
Cómo configurar adecuadamente un mousse para que funcionen los tres botones.

Ircll-COMO

Edith de la Torre, neptuno@vlc.servicom.es.
"Voy a escribir aquí todas mis penurias con el ircII. Primero serán un desastre, pero ya iré arreglándolas poco a poco. Todos tenemos problemillas con este IRC modo texto que en el fondo nos encanta, pero no encontramos info en castellano. Pues a ver si sale algo de aquí...".

JAZ-COMO

Bob Willmot, bwillmot@cnct.com.
Traductor: Iñaki Martínez Díez, **imd@redestb.es.**
Este COMO cubre la configuración y el uso de la unidad extraíble Iomega JAZ bajo Linux y las herramientas de software disponibles para ella.

Linux-sin-disco-duro-COMO

Robert Nemkin, buci@math.klte.hu.
Traducido por Rodolfo Pilas, **rodolfo@linux.org.uy**.
Este documento describe cómo configurar un sistema Linux sin disco.

Linux-con-raiz-NFS-COMO

Andreas Kostyrka, **andreas@ag.or.at.**
Traducido por Rodolfo Pilas, **rodolfo@linux.org.uy**.
Este Mini-COMO intenta explicar el procedimiento para configurar una estación de trabajo Linux sin disco, que monte la totalidad de su sistema de archivos vía NFS.

Bridge+Cortafuegos-Mini-COMO

Peter Breuer, ptb@it.uc3m.es.
Configuración de un sistema en el que coexista un cortafuegos con bridging de interfaces de red.

Gravis-Ultrasound-COMO

J. F. **Mammet, mammet@diva.univ-mlv.fr.**
Traducido por Iñaki Martínez Diez, **imd@redestb.es.**
Particularidades de configuración de la Gravis ULTRASOUND Plug n' Play bajo Linux.

Filtros-Impresión-COMO

Paco Andrés
Maquetador Linuxdoc-SGML: Antonio Ismael Olea González, **olea@iname.com**
2:345/108.9@fidonet.org.
Cómo elaborar filtros de impresión para Linux.

Fips-COMO

Arno Schaefer. Traducción de Juan José Montesinos Castellanos, **johncy@ibm.net.**
FIPS es un programa capaz de hacer nuevas particiones en el disco duro sin tener que borrar los datos. Es especialmente útil para hacer unafs particiónles para LINUX a partir de una única partición DOS.

En-Hora-COMO

Ron Bean, rbean@execp.com.
Traducido por Rodolfo Pilas, **rodolfo@linux.org.uy.**
Cómo configurar y mantener el reloj de su computadora en hora.

Discos-Grandes-COMO

Andries Brouwer, aeb@cwi.nl.
Traducción: Fco. J. Montilla, **pacopepe@insflug.org.**
Todo lo referente a geometrías de disco duro, así como al límite de los BIOSISO relacionados con particiones más allá de los 1024 cilindros.

DHCPd-Mini-COMO

Paul Makeev, mac@RoSprint.net.
Traducido por Pedro Pablo Fábrega, **mailto:pfabrega@arrakis.es.**
Brevísima reseña sobre la disponibilidad y la configuración de un servidor DHCPd en Linux.

Campana-Visual-COMO

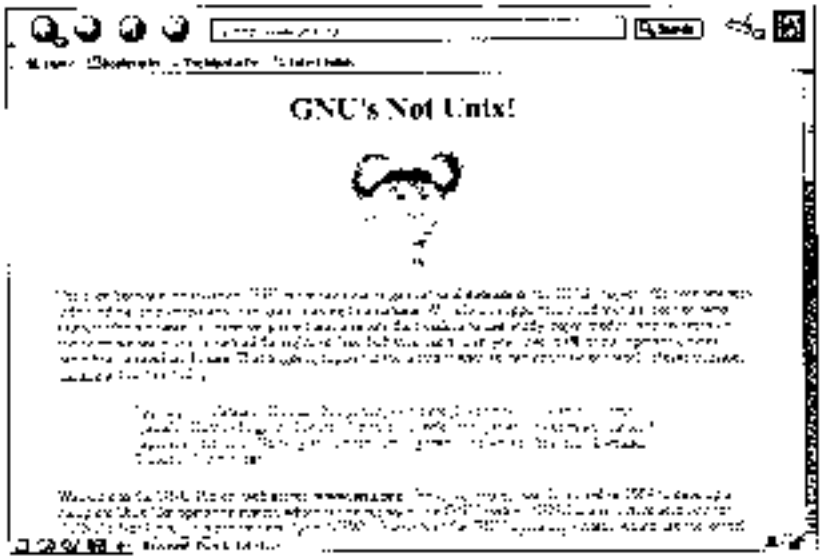
Alessandro Rubini, rubini@linux.it.
Traducido por Iosu Santurtún, **sorgina@sidam.es**.
Este documento muestra el uso de termcap para configurar una campana visual en el sistema propio y explica cómo deshabilitar las campanas sonoras al gusto de cada uno.

BogoMIPS-Mini-COMO

Wim C.A. van Dorst, baron@clifton.hobby.nl.
Traducido por Juan Carlos Durán García, **jcdg@hotmail.com.**
Este texto proporciona información sobre BogoMIPS, recopilada de varias fuentes mediante news y correo. Se puede obtener de varios servidores de archivos FTP de Linux en **linux/docs/HOWTO/mini/BogoMips.** *Se publicó un artículo en* Linux Journal, *en el número de enero de 1996.*

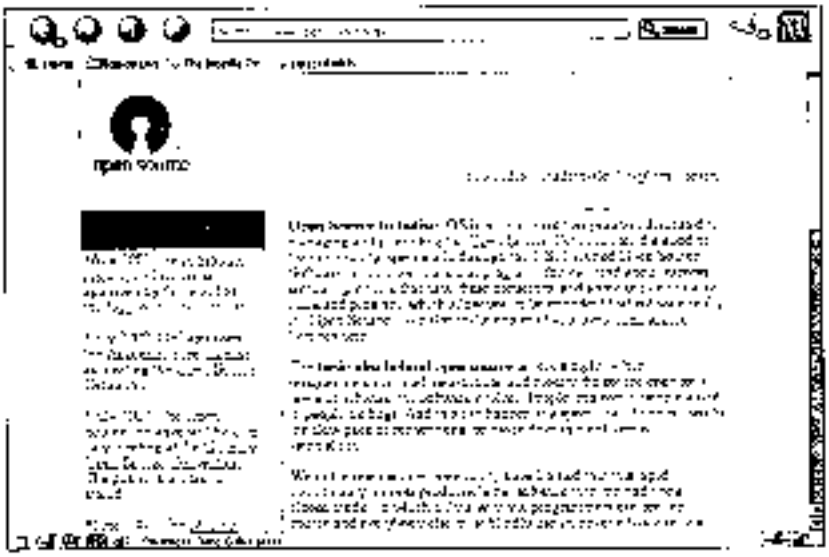
Guía de sitios web

GNU
www.gnu.org



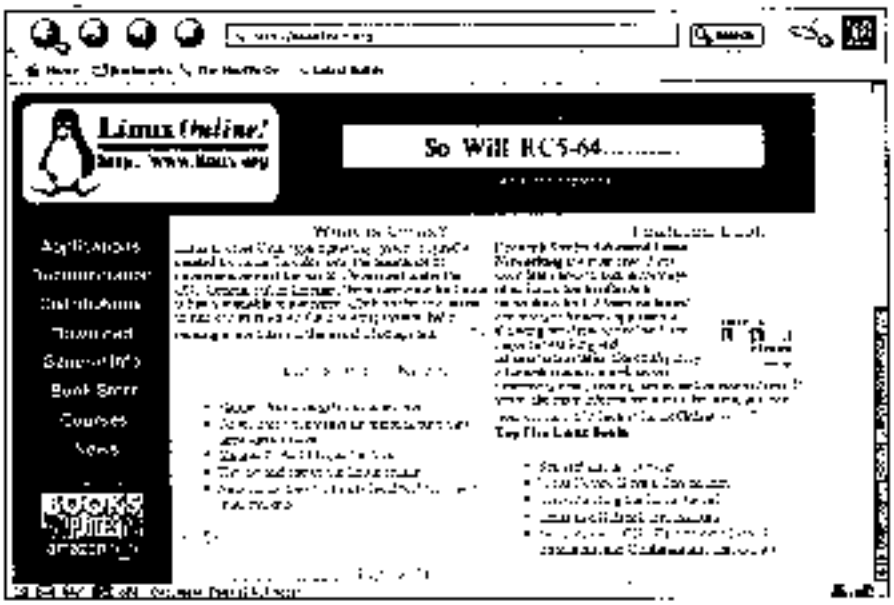
El sitio oficial del proyecto GNU/Linux. Aquí encontrarán mucha documentación (en diferentes idiomas), así como noticias y los sitios oficiales de los proyectos más importantes del sistema operativo GNU.

OpenSource
www.opensource.org



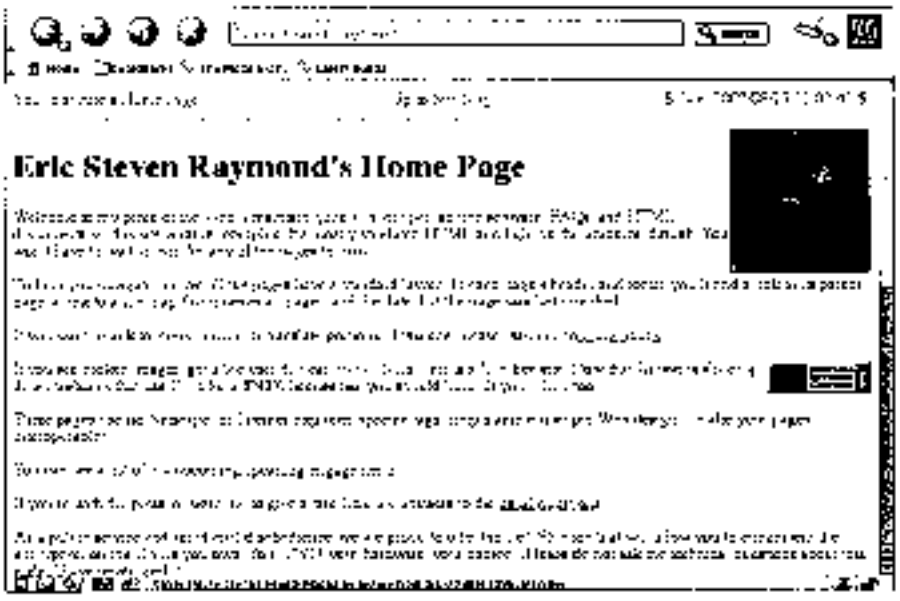
El sitio oficial de la iniciativa Open Source (código abierto).

LinuxOnline!
www.linux.org



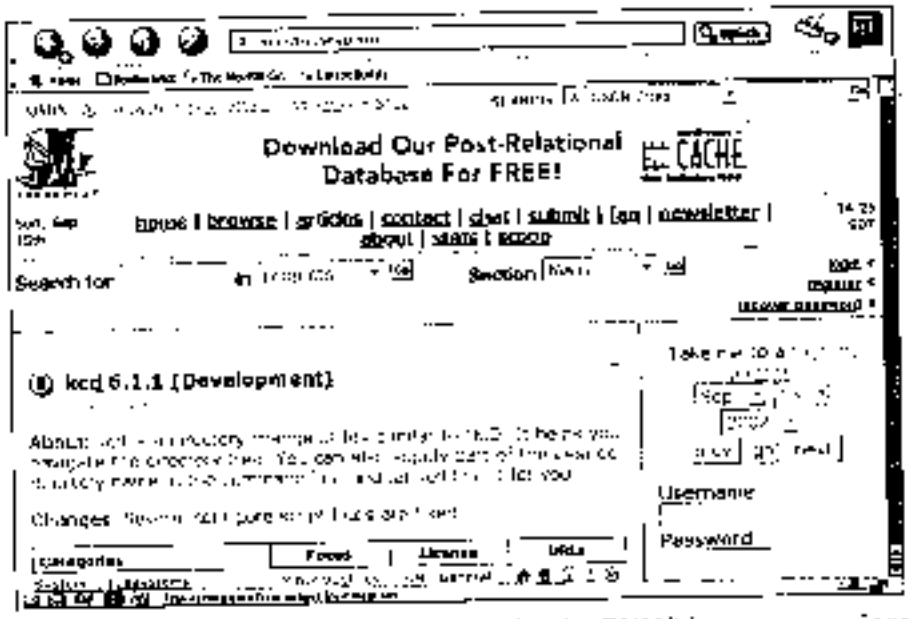
Un portal especialmente dedicado al sistema operativo GNU/Linux, con noticias, documentación y enlaces a otros sitios relacionados.

EricRaymond's Home Page
www.tuxedo.org/~esr



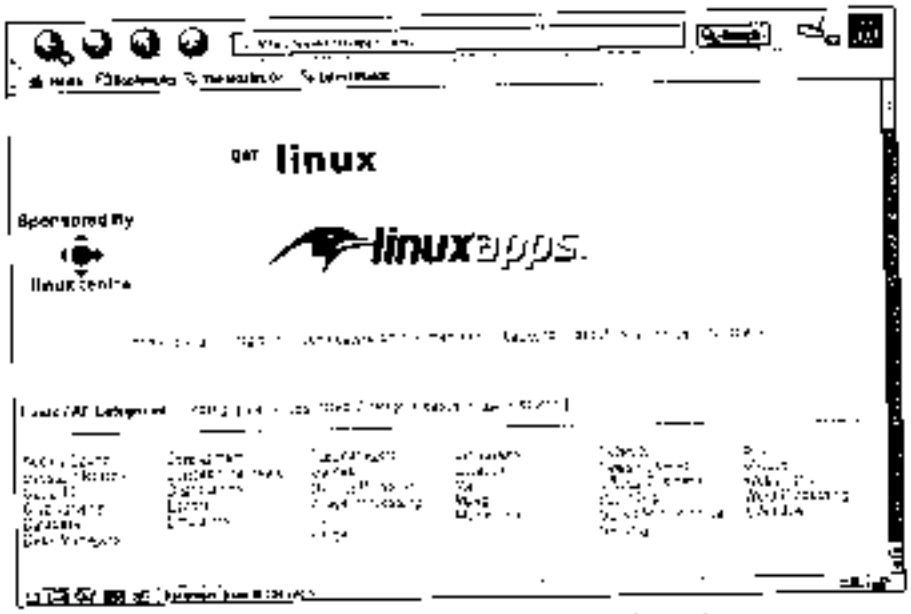
Eric Raymond es uno de los directivos de la iniciativa Open Source, y posee un sitio personal lleno de artículos, software de código abierto y mucho más. Eric Raymond es, además, el creador del cliente de e-mail Fetchmail.

Freshmeat
www.freshmeat.net



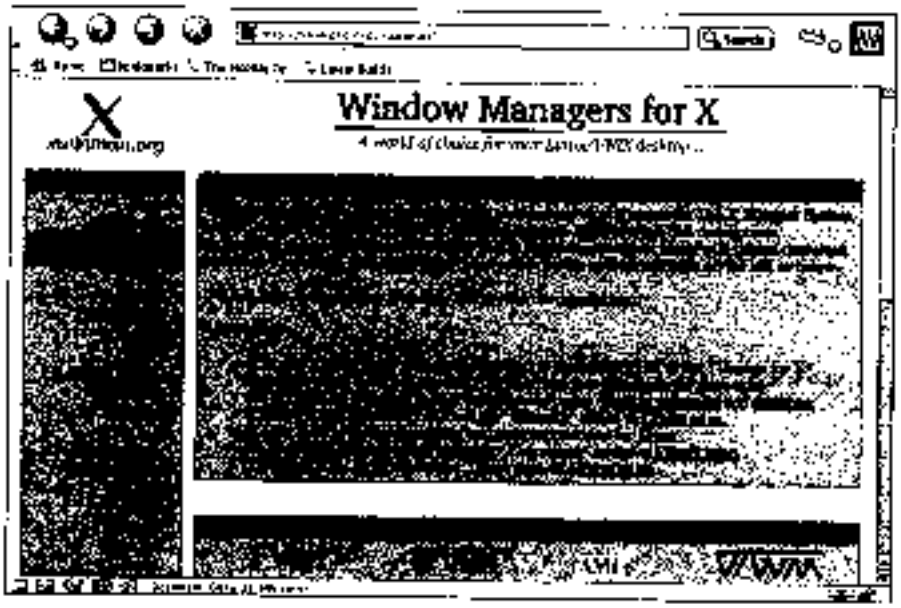
Freshmeat es el sitio obligado para todo usuario de GNU/Linux que quiera mantenerse al tanto de las últimas actualizaciones y lanzamientos de software.

Linuxapps
www.linuxapps.com



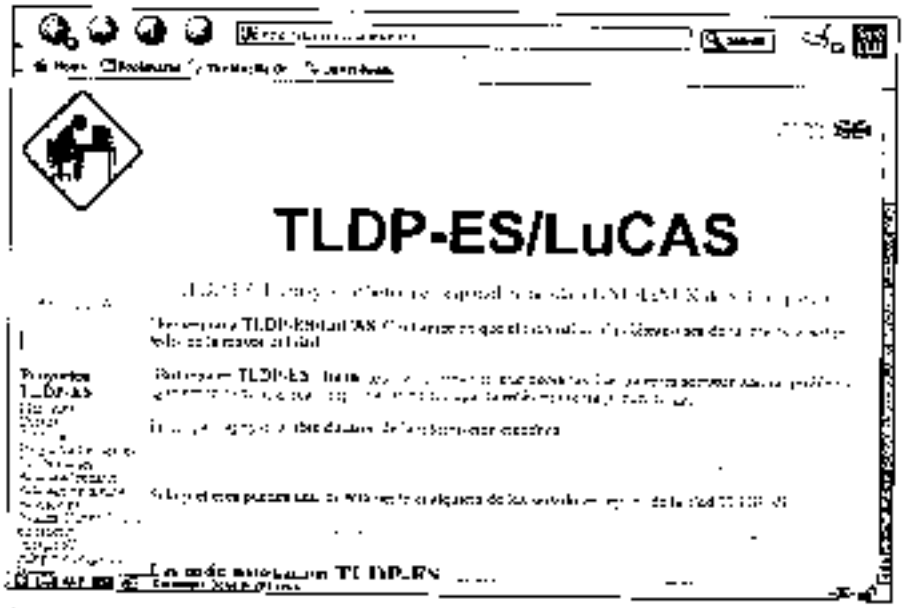
Otro sitio dedicado a las novedades en lanzamientos de software.

Xwinman
www.plig.org/xwinman/



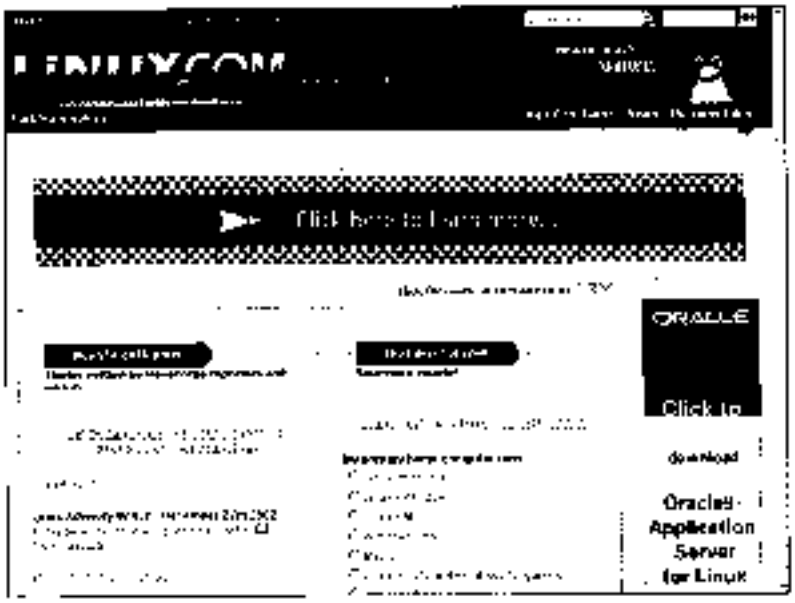
Un sitio personal dedicado a la revisión de los diferentes administradores de ventanas y escritorios para GNU/Linux.

ProyectoLuCAS
lucas.hispalinux.es/



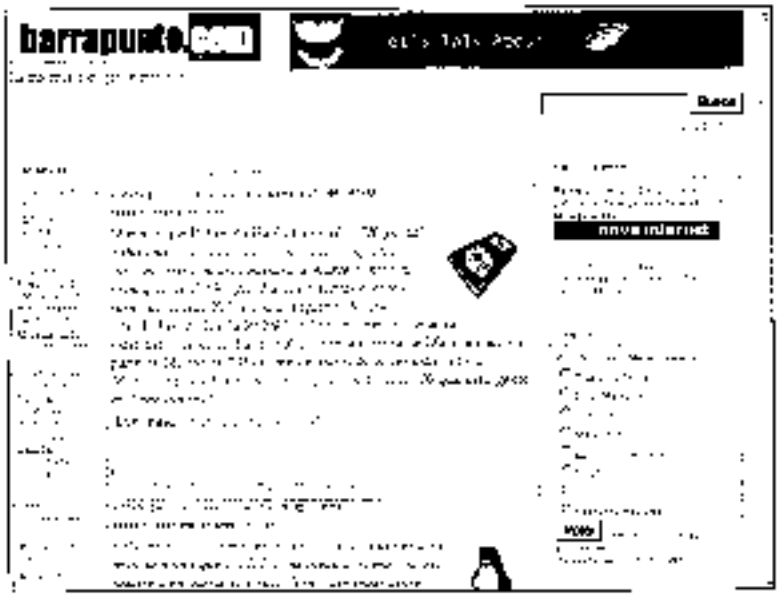
Según el sitio: "La mayor biblioteca en español dedicada a GNU/Linux de todo el planeta".

Linux.com
www.linux.com



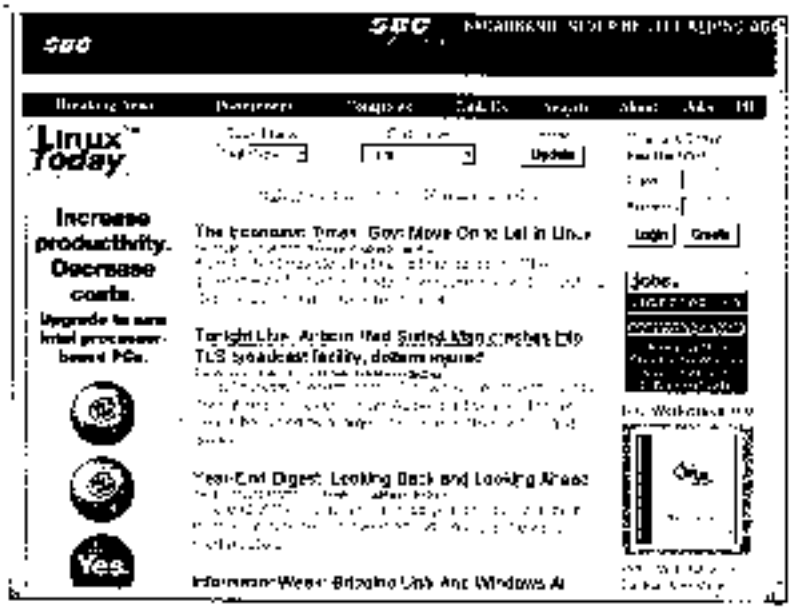
Sin dudas, se trata de uno de los portales más importantes sobre el tema. Entre su vasto contenido se destacan las noticias, la documentación, importantes informes y una gran cantidad de programas para Linux.

Barrapunto
www.barrapunto.com



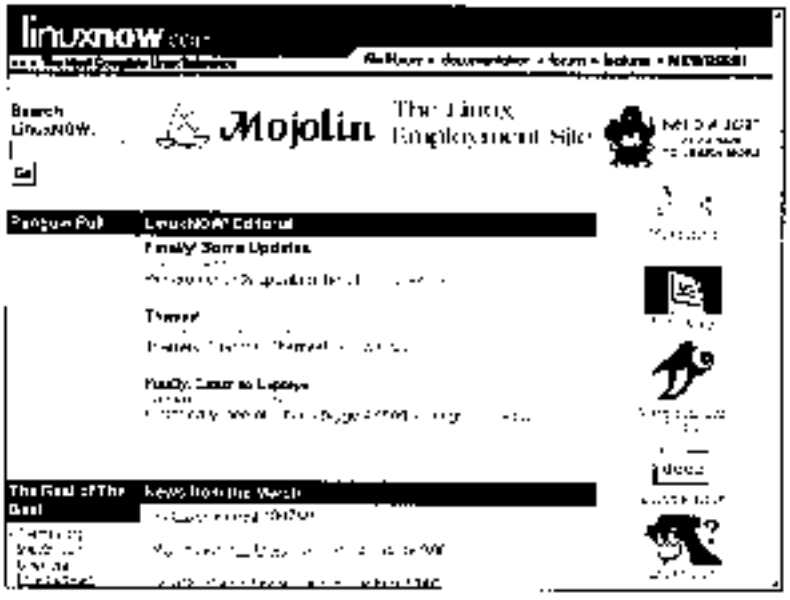
Completo sitio de noticias e información en castellano sobre el mundo de GNU Linux. Incluye links específicos para España, México y Argentina.

Linux Today
www.linuxtoday.com



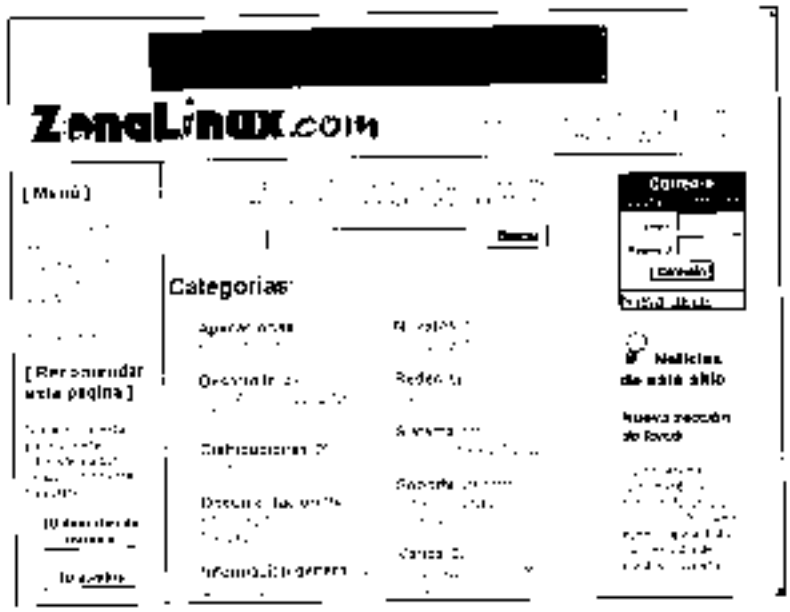
Noticias, noticias y más noticias sobre Linux. Es un referente imprescindible para todos aquellos que quieran seguir bien de cerca la actualidad del sistema operativo y todo su entorno.

LinuxNow
www.linuxnow.com



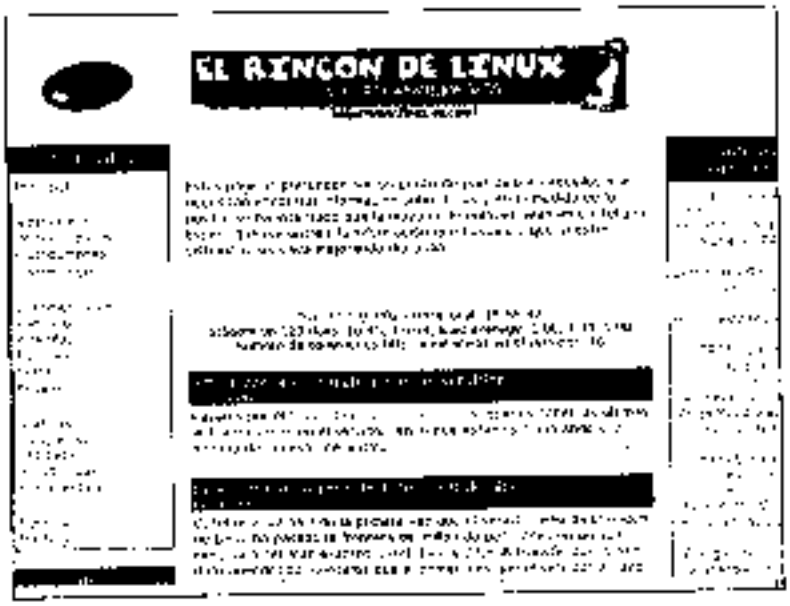
Otro portal con recursos, documentación, foros y una gran cantidad de links hacia diferentes sitios web relacionados con la temática.

ZonaLinux
www.zonalinux.com



Directorio al estilo "Yahoo", con gran cantidad de links relacionados, organizados por categorías. Entre las categorías se encuentran Aplicaciones, Distribuciones, Documentación, Novatos, Juegos, etc.

El Rincón de Linux
www.linux-es.com



El sitio de "Linux para Hispanohablantes" ofrece una gran cantidad de información y de software, así como una enorme cantidad de recursos sumamente útiles para los usuarios.

Glosario

Ada: lenguaje de programación que permite el desarrollo de aplicaciones multitarea.

Afterstep: uno de los Window Managers más populares en el mundo de Linux.

Alternarama: Servidor focalizado en la dirección **www.alternarama.com.ar**, que fue usado por el autor de este libro para montar varios servicios de red, entre los que se encuentran el sitio del mismo nombre, dedicado a las tecnologías alternativas. Lo interesante de este servidor es que inicialmente fue iniciado en un hardware 486 con 16MB de memoria RAM y un disco rígido de 1GB.

Ancho de banda: se llama así a la cantidad de datos que pueden ser transferidos a través de un canal determinado.

ANSI: *American National Standards Institute*. Organización sin fines de lucro dedicada a la estandarización de normas americanas. Además, es un estándar de transmisión de caracteres por sistemas de BBS.

Alias: método que proveen algunos shells para resumir un comando largo en uno corto.

Archie: sistema utilizado en Internet para la búsqueda de archivos en servidores FTP.

Archivo: es la unidad de almacenamiento en la mayoría de los sistemas operativos, incluida toda la gama UNIX. En Linux, los archivos pueden tener atributos que los convierten en archivos ejecutables, archivos de sólo lectura/escritura, o archivos que sólo pertenecen a un usuario en particular.

ARPANET: red WAN desarrollada por el Departamento de Defensa de los Estados Unidos. Es la precursora de la actual Internet.

ASCII: siglas de *American Standard Code for Information Interchange*. Es una tabla de símbolos con sus respectivos valores numéricos asignados.

Apache: el servidor web más utilizado en estos tiempos. Su dirección en Internet es: **www.apache.org**.

AppleTalk: sistema utilizado en máquinas Macintosh para compartir recursos en una red.

Awk: lenguaje interpretado.

Background: se llama así a los procesos que se ejecutan en otro plano del entorno. De esta manera se pueden ejecutar procesos en segundo plano trabajando independientemente y ejecutar un proceso en primer plano controlado por el usuario.

Backup: sistema utilizado para hacer copias de respaldo de la información del sistema.

Bandwidth: ver **Ancho de banda**.

Biff: programa de notificación de e-mails.

BIOS: siglas de *Basic Input Output System* (sistema básico de entrada y salida).

Bit: la más pequeña unidad de almacenamiento. Sólo puede almacenar un 0 o un 1.

BogoMIPS: la unidad que utiliza Linux para medir la velocidad de procesamiento en millones de instrucciones por segundo.

BOOT: proceso de booteo o inicio del sistema.

BootLoader: programa que se aloja en la MBR (*Master Boot Record*) y se encarga de cargar el núcleo del sistema operativo en memoria.

BSD: versión de UNIX desarrollada en la Universidad de Berkeley, California.

Browser: se llama así a los navegadores de páginas web en Internet.

Byte: unidad de almacenamiento de datos compuesta por 8 bits.

C: el lenguaje de programación utilizado para crear UNIX y Linux.

C++: lenguaje de programación basado en C, pero orientado a objetos.

Caldera: distribución comercial de Linux muy popular, destinada a empresas y organizaciones. Su sitio en Internet es: **www.caldera.com**.

CGI: siglas de *Common Gateway Interface*. Es un sistema para la conexión de programas con la Web.

Chat: conversación. Los usuarios de un sistema Linux pueden conversar mediante este sistema.

Consola: conjunto de teclado y pantalla conectados a una máquina UNIX. Linux permite manejar varias consolas virtuales en una misma máquina.

Clear: comando utilizado generalmente para borrar la pantalla.

Cluster: Se llama así a los sistemas de supercomputación, basados en varias computadoras que funcionan simúltaneamente. Las computadoras se reparten los procesos en base a cuantos recursos libres tiene cada una.

Comando: medio por el cual se le ordena una acción determinada al sistema operativo.

Compilar: se llama así al proceso de generación de un archivo ejecutable partiendo de su código fuente.

Consola: es el dispositivo mediante el cual el usuario envía y recibe los datos.

Consola virtual: Linux permite utilizar consolas virtuales. Éstas son máquinas virtuales independientes que corren sobre una sola máquina física y verdadera.

Cookie: galletita. Dato que un determinado sitio web le deja al navegador.

CPU: Unidad Central de Proceso.

Daemon: proceso que se ejecuta en el arranque del sistema y permanece en ejecución constante en segundo plano.

Debian: la única distribución de Linux que no es comercial. Considerada por algunos como la más segura y profesional. Esta distribución posee su propio formato de empaquetamiento de archivos llamado Debian Package Manager. Su sitio en Internet se encuentra en **www.debian.org**.

Demonio: ver **Daemon**.

Dial-up: se denomina así a las conexiones a Internet realizadas a través de un módem y una línea telefónica común.

Distribución: Linux se propaga en todo el mundo a través de distribuciones. Éstas son armadas por diferentes organizaciones que emplean distintos paquetes de utilidades, pero todas usan el mismo sistema (Linux). Ver: **Caldera, Red Hat, Slackware, Debian.**

Directorio: estructura utilizada por Linux para agrupar archivos.

DNS: sistema que permite transformar nombres a direcciones IP numéricas.

Domain: nombre identificador de un servidor. Los dominios pueden ser .com (comercial), .net (red), .mil (militar), .org (organización). Además, pueden contener dos letras identificadoras del país de ese servidor.

Dominio: ver **Domain.**

Download: término comúnmente utilizado para hacer referencia a la descarga de un archivo de un servidor.

E-mail: correo electrónico.

Eiffel: lenguaje de programación orientado a objetos. Su dirección en Internet es: **www.eiffel.tm.**

ELF: formato de archivos binario adoptado por Linux en su última etapa. Sucesor del a out.

EOF: siglas de *End Of File* o Fin de archivo.

Ext2: el formato del sistema de archivos de Linux.

FAQ: siglas de *Frequently Asked Questions* (cuestiones preguntadas frecuentemente). El mundo de Linux está lleno de este tipo de documentos, que informan de manera rápida y sencilla.

Finger: utilidad que permite obtener información de un usuario determinado, ya sea en la red local o en una WAN.

Freeware: software de libre distribución.

FTP: protocolo de Internet utilizado para la transferencia de archivos.

GNOME: proyecto abierto de desarrollo de un administrador de escritorios para el sistema de ventanas X.

GNU: siglas de "GNU is Not UNIX", el sistema operativo que tanto queremos.

Gopher: protocolo de Internet para la transferencia de datos en modo texto.

Hoy suplantado completamente por la Web.

GTK: librería para el desarrollo de aplicaciones en el sistema X.

Gurú: experto.

Gzip: una utilidad de libre distribución que permite comprimir archivos de Linux. Es la más utilizada para la compresión de aplicaciones de libre distribución en código fuente.

Hacker: persona aficionada a la resolución de problemas de distinta índole y a la búsqueda constante de información. Sus resultados siempre los comparte con los demás miembros de la comunidad. Para ser hacker, el aficionado deberá ser conocido y respetado por los demás ha-ckers.

HTML: siglas de *Hyper Text Markup Language*. Es el lenguaje utilizado para el desarrollo de páginas web.

Home: cada usuario de un sistema Linux es provisto de un directorio en el cual puede almacenar todos sus archivos de configuración y aplicaciones. Este directorio generalmente se encuentra ubicado en **/home/usuario**.

Home page: página de inicio. Muchas personas y organizaciones tienen su página informativa de inicio en Internet.

HOWTO: estructura de documentación de información muy utilizada en el mundo de Linux.

init: el primer proceso que ejecuta Linux. Éste, basándose en el archivo **/etc/inittab**, inicia todos los demás procesos configurados en el momento de inicialización del sistema.

Internet: la red WAN más grande del mundo.

IRC: protocolo de Internet utilizado para establecer conversaciones entre los usuarios de un servidor.

ISP: *Internet Service Provider*, proovedor de acceso a Internet.

JAVA: lenguaje de programación orientado a objetos de mucha utilización en Internet.

job: en Linux se llama así a los trabajos en ejecución.

KDE: uno de los administradores de escritorios para X más utilizados.

Kernel: es el núcleo del sistema operativo. Su principal función es gestionar los procesos, el manejo de la memoria, la entrada/salida y las unidades de almacenamiento.

ksh: Korn Shell; uno de los tantos shells para UNIX.

LAN: Red de Área Local.

LILO: el BootLoader más utilizado en Linux. Permite bootear casi cualquier sistema operativo.

Linux: kernel (núcleo) basado en UNIX desarrollado por Linus Torvalds durante principios de la década del '90.

Login: proceso de registro en una sesión Linux.

Logon: inicio de la sesión Linux.

Lynx: el navegador de modo texto más utilizado.

Mail: correo.

MIPS: siglas de Millones de Instrucciones Por Segundo. Unidad utilizada para medir la velocidad de procesamiento de un dispositivo.

MOSIX: Sistema que permite montar un cluster en GNU/Linux. Su característica principal es que consume muy pocos recursos, por lo que permite montar un clúster basado en computadoras de baja potencia.

Mozilla: Popular suite de herramientas para Internet. Incluye navegador, sistema de creación de páginas web, cliente de correo electrónico y más. Es multiplataforma.

Netscape: el browser gráfico más famoso de Linux. Funciona bajo el entorno de ventanas X. Su dirección en Internet es: **www.netscape.com**.

Newsgroup: grupo de mensajería localizado en la USENet. Un protocolo de transmisión de mensajes en Internet.

NNTP: el protocolo utilizado por el sistema USENet.

OSF: *Open Source Fundation*, Fundación de Código Abierto.

PGP: sigla de *Pretty Good Privacy*. El software de encriptación de mayor utilización.

Perl: sigla de *Practical Extraction and Report Language*. Uno de los lenguajes de script más famosos.

POP: protocolo de Internet utilizado para la descarga de mensajería desde un servidor.

PPP: siglas de *Point to Point Protocol* (Protocolo Punto a Punto), el protocolo de conexión a Internet utilizado para enlaces seriales.

Protocolo: lenguaje definido para la transmisión de información.

QWERTY: el modelo de teclado estándar americano.

Red Hat: una de las empresas dedicadas a la distribución de Linux en caja.

RFC: sigla de *Request For Comments*, un sistema de documentación utilizado en Internet desde sus comienzos, que trata los más diversos temas técnicos relacionados con la informática.

Root: raíz, en inglés. Este término se utiliza para nombrar el directorio principal de una partición. En los sistemas UNIX, el usuario que posee todos los privilegios se llama *root*.

Segmentation Fault (Error de segmento): Mensaje típico entre los sistemas UNIX (incluido GNU/Linux) que se produce cuando surge un error interno durante la ejecución de un programa (divisiones por

cero, referencias a espacios de memoria inexistentes, etc). Generalmente guardan el estado actual e la memoria en un archivo llamado core.

Sendmail: el gestor de mensajeria de mayor popularidad en los sistemas basados en UNIX.

Server: servidor, en inglés. Se llama servidor al equipo que provee de información o recursos a diferentes clientes.

SGML: sigla de *Standard Generalized Markup Language*. Formato de documentación utilizado por algunas organizaciones de documentación de aplicaciones Linux.

Slackware: Una de las distribuciones de Linux más seguras, rápidas y difundidas.

Shell: programa que permite la comunicación entre el usuario y el sistema operativo mediante el ingreso de comandos.

SMTP: sigla de *Simple Mail Transport Protocol*. Protocolo utilizado para el envio de mensajeria.

TAR: utilidad de gran difusión en los sistemas basados en UNIX para el empaquetamiento de varios archivos.

Terminal: equipamiento generalmente basado en pantalla y teclado que se conecta a un servidor.

TeX: un sistema de formateo de documentos.

TCP/IP: siglas de *Transmission Control Protocol/Internet Protocol*. El protocolo estándar de Internet.

Telnet: protocolo de Internet utilizado para el acceso remoto a un servidor.

UNIX: sistema operativo multiusuario y multitarea. Con él comenzó todo.

URL: sigla de *Uniform Resource Locator*. Sistema de direcciones utilizado en diferentes protocolos de Internet.

UseNet: sistema de intercambio de mensajeria basado en el protocolo NNTP.

WAN: sigla de *Wide Area Network*, Red de Área Amplia.

WWW: sigla de *World Wide Web*. Sistema de transferencia de hipertextos en Internet. Utiliza el protocolo HTTP.

X11: sistema de ventanas utilizado en sistemas UNIX.

Xfree86: la implementación del sistema de ventanas X11 más utilizada en Linux.

Bibliografía recomendada

A continuación presentamos una serie de libros recomendados para todos aquellos que, o bien quieran investigar y profundizar en la materia, o que deseen perfeccionarse analizando concretamente alguno de los temas específicos o aplicaciones del sistema operativo. Como se puede apreciar, entre los temas concretos se incluye bibliografía sobre seguridad y sobre la implementación de sitios web bajo Linux, precisamente del mismo autor de esta obra.

UNIX Sistema V Versión 4 (2da. edición), Rosen-Rosinski-Farber-Host. Editorial McGraw-Hill Interamericana.

Linux Máxima Seguridad, Anónimo. Editorial Prentice Hall.

Red Hat Linux 7.1 Secrets, Naba Barkakati. John Wiley & Sons.

Sitios Web bajo Linux, Facundo Arena. MP Ediciones.

Linux Avanzado, Facundo Arena. MP Ediciones.

Índice temático

MANUALES

USERS Linux

Este libro esta dirigido a todos aquellos que quieran relacionarse (o ya tengan relación) con el sistema operativo GNU/Linux. Aquí se tratan los más diversos temas, de forma que quien no conoce absolutamente nada acerca de este sistema puede comenzar a dar sus primeros pasos con él. Por su parte, los usuarios que ya tienen experiencia pueden encontrar conceptos útiles y formas de desarrollar soluciones a los diferentes problemas que se presentan habitualmente en los sistemas informáticos.

Los primeros capítulos son introductorios, y sirven para explicar la instalación y el uso básico de Linux. A partir de Capítulo 3 se presenta toda la información necesaria para manejar el sistema a fondo, principalmente desde el modo texto. En los últimos capítulos se incluye información para usuarios avanzados, así como también, direcciones para obtener más recursos en Internet.

SOFTWARE RECOMENDADO

WEB

En este sitio encontrará una gran variedad de recursos y software recomendado, que le servirán como complemento al contenido de los capítulos, brindará la posibilidad de estar al contacto con los autores, y de participar del foro de lectores, en donde podrá intercambiar opiniones y experiencias.

Para mayor información sobre el libro comuníquese con nuestro Servicio de Atención al Lector

EN ARGENTINA [illegible] (C1064ABF) Ciudad de Buenos Aires
[illegible]

EN MÉXICO Av. Río Churubusco No. 361, Xoco, Delegación Benito Juárez, C.P. 03330, México D.F. Teléfono: 5238 0230

RESTO DE AMÉRICA [illegible]

NIVEL DE USUARIO

PRINCIPIANTE | INTERMEDIO | AVANZADO | EXPERTO

ISBN 987-526-120-2

9 789875 261204